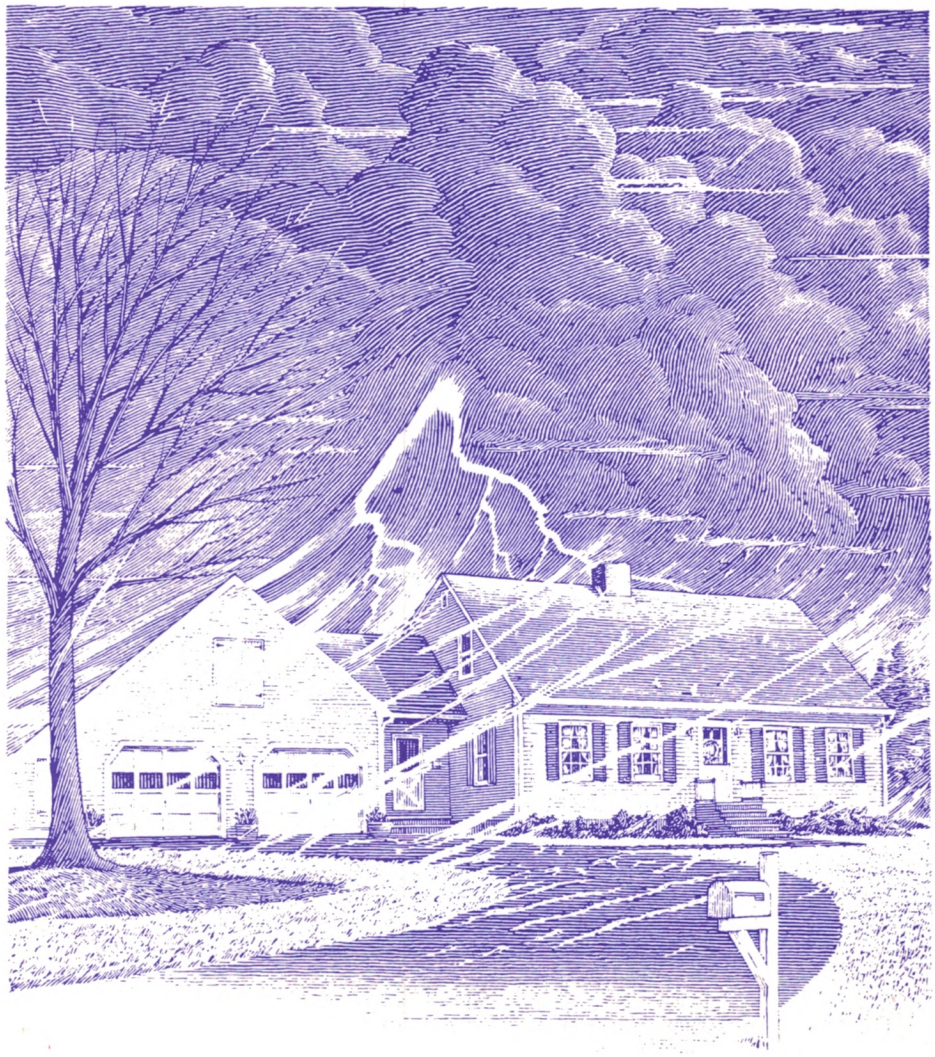
THE MISOSYS QUARTERLY

In this issue:

- ☞ The Soltoffs' have a 7 pound 12 ounce boy!
- ☞ Reading NEWDOS/80 disks, by David Goben
- ☞ An LB Archival Utility, by Daniel L. Srebnick
- ☞ Popup Application Window, by Roy Soltoff
- ☞ XMODEM in C, by David B. Lamkins
- ☞ Getting into computer math, By Roy Soltoff
- ☞ and **An index to *THE MISOSYS QUARTERLY*, Volume I**



THIS IS

WHY YOU

NEED AN

U.P.S.

Volume III.i **$10** **Summer 1988**

TRS-80 Model I/III/4 Language Software from MISOSYS

Z80 RELocatable Assembler

MRAS [DOS 6.x M-21-083] [LDOS 5.x M20-083]

An advanced Z80 assembly package for the programmer who wants a powerful and flexible development system. It includes a macro assembler which generates either **relocatable object code** modules or CMD files directly, a linker, a librarian, a full-screen text editor, a utility for converting to/from line-numbered files, and a cross reference tool for directly generated CMD files.

MRAS generates M80 compatible /REL files. Supports REPT, IRP, and IRPC macros; nested includes, and a full range of nested conditionals. It has flexible output redirection of listing and symbol table.

MLINK supports virtual memory bit-stream buffering, REL and IRL library searching, zero disk space for DEFS in DSEGs and COMMONs, generation of program overlays, special link items: 0-3, 5-7, 9-11, 13-15.

Includes **MLIB**, our REL module librarian, and our **SAID** advanced **full screen text editor** which can be used to generate your assembler source code, C-language source code, or edit any type of ASCII file.

BASIC Compiler

EnhComp [DOS 6 M-21-072] [LDOS 5 M-20-072]

Released in 1986 and reviewed in the March 1987 issue of *80 Microcomputing* and October 1987 issue of *COMPUTER SHOPPER*. EnhComp has lots of great features; handles the bulk of Model III Microsoft BASIC and supports additional commands and functions. Standard is floating point with both **single and double precision** functions; random file access ("X" mode for reclens to 32767), turtle graphics, pixel graphics, keyed array sort, multi-lined functions, user commands, REPEAT-UNTIL, line labels, and more.

A *supervisor* program automates the edit-compile-test phases inherent when using compilers; this makes using EnhComp almost as easy to use as your BASIC interpreter.

With its built-in **Z80 assembler**, you can **easily** create hybrid programs of BASIC statements and in-line assembly code which **completely eliminate** contorted string packing and DATA statement high-memory module techniques for your BASIC program to access a machine code module.

You'll have to edit *existing* BASIC programs, but the power and completeness of EnhComp make that an easy task.

RATFOR Compiler

RATFOR-M4 #M-21-073

RATFOR compiles to FORTRAN; is **fully structured** facilitating modification and debugging; comments are simpler and more versatile than in FORTRAN, simplifying self-documentation. This allows changes without the subsequent debugging tolerated when modifying FORTRAN. RATFOR *compiles* source code to an *object* of FORTRAN; use your existing FORTRAN compiler to convert this to executable.

RATFOR is an excellent language for general purpose use, but it is vastly superior to FORTRAN when working with a large number of modules without documentation, as is necessary when producing very large programs.

Extensions supported include the "arith" macro to perform binary arithmetic operations, read and print macros for short form READ and PRINT, and support of any valid FORTRAN expression for "switch" and "case" operands.

This package includes the language translator, a *batch* file to automate compilation, a language Reference Manual, an Installation Manual, application programs in source code on disk, and our LED text editor for source code preparation.

Z80 Assembler

EDAS [DOS 6 M-21-082] [LDOS 5.x M-20-082]

This powerful combined disk-based **line editor** and Z80 **macro assembler** assembles from nested source files or memory buffer; nested conditionals with ten pseudo-ops, nested MACROs with parameters both positional and by keyword, cross reference listings; and a separate **full screen text editor**.

If you are writing system software, support software, applications - big or small, EDAS will provide the power to make your job easier, faster, and more worthwhile.

Full C Compiler

MC [DOS 6.x M-21-064] [LDOS 5.x M-20-064]

If you are looking for a **full C compiler**, look no further. If you are looking for a well stocked UNIX System V standard library, look no further. **MC**, reviewed in the January 1987 issue of *80 MICROCOMPUTING*, is a complete C compiler which adheres to the standards established by Kernighan and Ritchie. The library of functions is extensive and System V compatible. The compiler generates Z80 relocatable macro assembler code (M80 or our MRAS). The libraries are files of relocatable object modules. **MC** is a full-featured compiler for the discriminating programmer!

MC supports command line *I/O redirection* for compiled programs, *wild-card* file specifications, parsing for UNIX "." extensions in file specifications, *overlay* support (requires MRAS), a full pre-processor, lots of options, and is designed for the programmer wishing the ultimate in C compilers. The package is supplied with the compiler, pre-processor, an optimizer, assembler macro files, C libraries, a Job Control Language file, the header files, and a 400+ page user manual. MC requires the use of either M-80 or MRAS (available separately), 2 disk drives, and upper/lower case.

Z80 Disassembler

DSMBLR [DOS 6 M-31-053] [LDOS 5.x M-30-053]

Provides direct disassembly from CMD disk files, automatic partitioning of output disk files, data screening for non-code regions, and ***full label generation***. It even generates the ORGs and END statement - the complete ball of wax. You will find that the use of this disassembler - *even by a beginning assembly language programmer* - will be paying handsome rewards with the ease of its use and clarity of the documentation. It's a professional tool for your use.

The disassembler allows you to build a *screening data file* telling what segments of the program are to be interpreted as data regions. You enter the addresses of the "segments" after analyzing the target program's disassembly.

Output to **DISK** produces a file suitable for MRAS/EDAS and is automatically segmented into manageable file sizes.

Special Sale Prices

Product	Specify Model	Sale$	S&H
DSMBLR	~~$34.95~~	$20.98	$2
EDAS	~~$74.95~~	$44.98	$4
EnhComp	~~$99.95~~	$59.98	$4
HartFORTH	~~$59.95~~	$35.98	$3
MC	~~$124.95~~	$74.98	$5
MRAS	~~$89.95~~	$53.98	$4
RATFOR	~~$99.95~~	$59.98	$5
UNREL	~~$49.95~~	$29.98	$2

Sale price is continued through 12/31/88: You must include TMQ coupon. U.S. Add S&H. Canada use U.S. + $1. Foreign S&H use U.S. * 3. M/C & VISA accepted. No COD's for this sale. 30-day refund on product if not acceptable.

MISOSYS, Inc.
PO Box 239
Sterling, VA 22170-0239

REL Disassembler

UNREL [T80 M-30-054] [CPM M-32-054]

Decodes an M80-type relocatable object module and outputs an MRAS/M80 assembler source file. We bundle in **SPLITLIB** to split a library into separate modules and **DECODREL** to display the *bit stream* of a REL file.

UNREL assumes anything in a code segment is code, and anything in a data segment is data. It supports special link items: 0-3, 5-7, 9-11, 13-15.

UNREL should be the perfect professional assembler's tool for your bag of tricks.

FORTH Compiler

HartFORTH [DOS 6 M-21-071][DOS 5 M-20-071]

HartFORTH is a ***full FORTH*** that conforms to the **79-STANDARD**. The Model I/III version is an indirect threaded version; the DOS 6 version is a direct threaded implementation providing greater execution speed of 10%-40% depending on the details of the actual program. The kernal contains some additional useful words and utilities which turn HartFORTH into a full-fledged development system.

HartFORTH is designed to ***run under an operating system*** which is totally transparent to the programmer or user. The virtual Memory that it accesses for storage and retrieval purposes is a normal DOS file that is requested by the FORTH system when it is first entered. Doing this has several advantages in that it provides for FORTH files to be used in other language application programs and vice-versa. Enhancements have been built into the kernal in the form of functions to call the operating system file handling routines so that other files may be created or accessed.

HartFORTH supports double length integers, string handling, cursor manipulation, graphics, random numbers, and floating point.

Take any ONE language product at 40% off!

THE MISOSYS QUARTERLY

Volume III.i **Summer 1988**

Table of Contents

The Blurb 2
New Product Announcements 6
Family Update 6

Letters to the Editor 8

DOS Subjects 21
LDOS™ and Model 3 Information 21
LS-DOS™ and Model 4 Information 22
MS-DOS™ Information 25

Roy's Technical Corner 27
Getting into computer math 27

Applications for the User 31
Reading NEWDOS/80 disks, by David Goben 31
XMODEM in C, by David B. Lamkins 35
An LB Archival utility, by Daniel Srebnick 55 38

The Programmer's Corner 41
Popup Application Window, by Roy Soltoff 41

MISOSYS Products' Tidbits 52
MC C-compiler 52
EnhComp 55
Little Brother Data Manager 56
PRO-WAM Windows & Applications 57

Data sheets on UPS/SPS 59

The Hardware Corner 63
International keyboards 63
Hard drives 64
Notes on the XLR8er 66
PCB Hardware tips 71

Index to TMQ Volume I 72

Copyright © 1988 by MISOSYS, Inc., All rights reserved
PO Box 239, Sterling, VA 22170-0239
703-450-4181

The Blurb by Roy Soltoff

Points to Ponder

Since this may very well be the first column you turn to when you open up your issue of TMQ, let me first apologize in advance if you feel this issue is a little rough behind the edges. I wanted to make sure that this issue was completed and off to our printer prior to going on vacation. *The Blurb* is the last piece I generate for an issue. As I write this, it is Wednesday August 3rd, and the camera ready copy for the entire issue must be ready by tomorrow afternoon. Since I can't preprint the remainder of the magazine because the page numbers will depend on how long *The Blurb* is, final printing takes place tomorrow morning. That's a good reason to have shifted over to a Laser printer which takes no more than 15 minutes to print the entire issue.

I normally would have had the issue ready for the printer by the 19th, or 22nd; thus, I have pulled up the deadline by two weeks. That did not allow me the time to spend a week browsing through all of the archived messages captured from the MISOSYS forum on Compuserve (PCS49). I usually extract dialog which I feel is of importance to TMQ readers. That content usually represents about 15-20 pages of material for TMQ. So this issue will be a little slimmer; it in no way is a trend. It gave me the opportunity of inserting the index to TMQ Volume I which was prepared by Elmar Von Muralt. You'll find that index at the tail end of this TMQ. I expect to insert the index to Volume II in a later issue.

Both Brenda and I have been extremely busy over the past few months. You recollect from the last issue that Brenda broke her ankle some months back. That put a crimp on what I was able to get accomplished since she was off work for two-three weeks. With the birth of Benjamin in June (see Brenda's *Family Corner* at the end of *The Blurb*), Brenda was again away from MISOSYS for almost a month. Believe me, she was really anxious to get back to work. Since MISOSYS sent out an 11,000 piece mailing in Mid-May, sent out 850 Quarterlies at the end of May, sent out over 400 TMQ renewal notices at the beginning of June, and took over the sale and support of LS-DOS 6.3 June 1st, during most of June, I was able to nothing but just take care of orders. My desk, previously backlogged with work, now looks like a mountain. I am probably two or more months behind in handling correspondence. I am now back to the position where some mail will not get answered. So if you sent in a letter expecting an immediate reply, it just won't happen.

I had expected in July to release a serial driver package for the XLR8er; however that project is behind schedule. I actually started on it. In fact, I felt that I would start out by adapting my disassembler and my MRAS assembler to support the 64180 (or Z180) OP codes. I have completed the disassembler but I have yet to test out the Model 3 version. So I will hold off from making any formal announcement until next issue. I then modified MRAS to support the 64180 OP codes as well (via a switch option). Before I released a new version of MRAS, I wanted to enhance the MLINK linker to support the chain address special link item required to enable it to link Microsoft FORTRAN-generated REL modules. That's where I ran into a small snag. Handling the link item was very easy; the chain address is just a subset of the chain external link item which was already supported by MLINK. Unfortunately, since the FORTRAN compiler is a one pass compiler, it doesn't generate the size of each segment (code, data, common) until the end of the code stream. The current architecture of MLINK establishes a memory block for each module segment used to store bytes for the segment. This requires that the segment size link item must precede any bytes which are part of the segment. To handle FORTRAN linking, MLINK needs to either incorporate a sophisticated memory management scheme, or read the input module twice to establish the segment sizes. I haven't yet decided which way to go. I suspect that I may decide on the latter and provide a preliminary pass under switch control. MLINK could easily detect the condition of sensing a segment byte before the segment was sized. So this is why I can't formally announce a new MRAS release. I also had not planned to alter the EDAS or PRO-CREATE assemblers, but since the code changes have already been done for MRAS, altering EDAS will only require me to re-groom the source code files to incorporate all of the patches which have been applied, I now expect to add the 64180 support to EDAS/PRO-CREATE.

As far as the XLR8er driver is concerned, that ran into a few snags. Remember, folks, that the driver support was supposed to be provided by H.I.Tech many months ago. MISOSYS never had the responsibility for that. However, since we have taken over the distribution of XLR8er boards, we have been forced to do the work. First, I discovered that the power cables supplied by H.I.Tech were incorrectly assembled. Next, it appears that the CTS status line uses a voltage level for a logic high which is of opposite sense to that normally used in the TRS-80. Furthermore, in order to be of use as a serial port, the 64180 Asynchronous Serial Communication Interface (ASCI) needs to be operated in a mode which at least uses the receive character interrupt. Since the internal interrupt controller for 64180 enhanced functions operates similar to standard Z80 interrupt mode 2, it is more difficult to simultaneously support

the ASCI interrupts as well as the standard mode 1 interrupts of the model 4 under the current architecture of the DOS without some very tricky code. The internal interrupt handler must be coded to reside in low memory, must include a 32-byte region at a 32-byte boundary for internal interrupt vectors, and must support memory management to switch in memory bank 0 in order to handle a serial driver anywhere in the 64K address space. I need to do some more research and soul searching to decide on the final plan of attack. The simplified driver I have written which doesn't use interrupts is just not effective for general work. I expect to thoroughly digest the HIBANKS, FIXBANKS, and FIXALL modules to explore a method of integrating the internal interrupt handling within the XLR8er interfacing software. So the serial driver is on the back burner until other matters are first dealt with.

Now for some good news. I came across an interesting article in *Computer Reseller News*, dated June 20, 1988. The article, Computer Execs' Portfolios Rebound, covered the Microcomputer executives' stock holdings and their current value for the top 165 executives. The micro computer industry is not that old, yet it has made multi-$millions in paper profits for some folks, and even one billionaire. It's interesting that Jon Shirley, an exec at Tandy a few years back who moved over to Microsoft, now has stock holdings worth $45,507,774. Here's the list of the top 10 Software stockholders.

1. William Gates, Microsoft	$1,296,620,942
2. Lawrence Ellison, Oracle	$153,599,909
3. Charles Wang, Comp.Assoc.	$133,255,237
4. Paul Brainerd, Aldus	$75,000,000
5. Anthony Wang, Comp.Assoc.	$70,847,700
6. John Walker, Autodesk	$66,554,019
7, Jon Shirley, Microsoft	$45,507,774
8. Jim Manzi, Lotus	$32,553,585
9. Edward Esber, Ashton Tate	$29,268,000
10. John Warnock, Adobe	$18,625,840

I came across an article in *Government Computer News*, June 24, 1988, headlined, "1999 Could well be a bonanza year for consultants". The thrust of the article was, you guessed it, all of the software programs from micros to mainframes, which can't deal with dating past 1999. Personally, I would like to be apprised of any computer in use after 1999 which is currently being used today.

My next comments are directed to Jim Whitaker, of SYDTRUG, down in N.S.W. Australia. Yes, Jim, I read your blurb in the July issue of SYDTRUG News. I do expect to totally unbundle the components of our proposed hard drive package. Thus, anyone will be able to purchase the drive, controller, cable, drive case, host adaptor with software, etc. individually or all together. I also double checked the power supply in the drive case and it does have an internal switchover for 115V/230V AC power. It comes with the typical removable power cord with the standard American-type three-prong plug. If that helps our off-shore customers, you folks may only need to revise the AC power cable with a different plug. I cover a little more detail on that hard drive package later.

My final comments are directed to Gordon Collins, of NATGUG, a user group over in England. No Gordon, I won't "shoot you at dawn" for reprinting some brief excerpts of TMQ in the *NATGUG NEWS*. Certainly informational items are free to reprint in any club newsletter if it helps the cause of communicating valuable information among TRS-80 users. I just don't want wholesale re-printing of articles.

I thoroughly enjoyed reading John Coyne's article on "Moving Superlog 4 into the XLR8 Board" in your August 1988 issue, as Gordon uncovered the problem of using simultaneously, both Superlog and PRO-WAM. That problem was reported to me quite some time ago, but not being a Superlog user (or owner), I never had the time or experience with Superlog to investigate that operating conflict. John uncovered the conflict to be excessive use of the system stack.

Let me put a little into perspective. When TRSDOS 6.0 was released, the DOS had a 256-byte stack. Release 6.0.1 eliminated the restriction on a program stack being below X'F400' by adding stack management support code to the video and keyboard device handlers to utilize a second system stack if the current stack pointer was above that limit during a keyboard or video device I/O request. The limit, by the way, stems from the use of the X'F400'-X'FFFF' address range for imaging the keyboard and video memory maps. Since the DOS had no unused low-memory region for a second stack, it split the then 256-byte stack into two 128-byte segments. When 6.1.0 was released by LSI, the low-memory requirements for added DOS routines forced the system stack to be reduced to 128 bytes. LSI originally only provided the "second" stack to be 32-bytes below the first. Unfortunately for some programs, that wasn't adequate so a patch was applied to split the 128-byte stack region into two areas of 64 bytes each. That's the famous FIXBOOT patch with PRO-WAM.

John has now found that even 64 bytes of system stack are insufficient when both Superlog and PRO-WAM are used simultaneously. John prepared a patch to Superlog to aid in dealing with that problem. Another solution may be to use the free low memory space below 1300H (the I/O driver region) if any exists in a particular configuration. I would suggest that if there were, say, 64 bytes available, just patch the stack switch instruction to point to the top of that 64-byte area, and the same for the SPOOLer patch. Then adjust the header pointer in the immediately preceding module to point to the byte following that 64-byte block (the pointer would have been pointing to the first free byte which you are now using). This technique essentially extends the last module in the low memory region by 64 bytes and avoids the requirement for another module header. I used that technique in my Alpha Technology memory board patches.

MISOSYS Hot List

This list represents what products in our catalogs have been the most popular in terms of sales units. The list excludes TMQ subscriptions and DISK NOTES. I have added LS-DOS 6.3 to the list effective June 1988. Since it will probably always be at the top of the list, I'm not sure it provides much additional information. Let me know what you think. The current month is July 1988; prior 3 months is Apr-Jun 1988; and prior 12 months is Jul87-Jun88.

MISOSYS HOT List

Current Month	Prior 3 months	Prior 12 months
LS-DOS 6.3	LDOS 5.3	LDOS 5.3
LDOS 5.3	*The Source*	PRO-WAM
The Source	PRO-WAM	Little Brother
MRAS	Little Brother	*The Source*
Little Brother	MRAS	diskDISK
PRO-WAM	DSMBLR	RSHARD

TMQ Schedule

Our target for mailing the *THE MISOSYS QUARTERLY* is the last week of the respective month as follows: Winter issue in February, Spring issue in May, Summer issue in August, and Fall issue in November. This schedule may place your TMQ late in the season based on the cover date; however, it follows from the mailing of issue I.i on August 19th, 1986.

Because of our vacation from August 6th to the 14th, this issue was prepared early; it was off to our printer on August 4th. As I write this, I expect to place this issue in the mail by Wednesday August 17th. That pulls up the mailing date about 1.5 weeks. We hope to continue to pull up the mailing date of succeeding issues until the target delivery is close to the beginning of the season.

The TMQ coupon has a field for entering your receipt date. If you are returning the coupon, please note the receipt date on the coupon and get it back to us. Unfortunately for the last issue, a few folks recorded "yes" instead of the date of receipt. If you send me back the coupon, I certainly know that you got the issue. Boy I really have to make things simple; the DATE OF RECEIPT, folks. I have not had the time to analyze the date of receipt recorded by those readers who returned their coupon. As noted, I've been swamped with work.

I am considering applying for a **Second Class** mailing permit for *THE MISOSYS QUARTERLY*. If I do pursue that avenue, it will actually make the distribution far easier for us. That's because I can then make do with one imprint, worldwide, and all postage costs are paid with a lump sum check to the USPS, rather than individual metering. I also would have my printer do the labeling, bundling, and trek to the post office. I would move to a Cheshire label - the kind you see on all high volume mailings. Thus, if I went to Second Class, once I passed the camera ready copy over to the printer, I would have no further involvement. That sounds "peachy keen" to me.

Domestic Second Class rates are a little lower than Third Class, but foreign rates are identical; a magazine still is subject to the AO Air Printed Matter rate, even if mailed under a Second Class mail permit. But all subscribers will have to move to Second Class; I wouldn't want to have mixed classes. Since the decision has yet to be made, I cannot advise you of any impact. But I thought you should know. Most magazines are mailed at the Second Class rates.

I also note that 3rd class postal rates went up over 25%; 1st class and international mailing rates took a steep increase in excess of 17% as well. For instance, mailing a ten-ounce TMQ used to be $1.75, $3.58, and $4.72 for 1st class US, AO Air to Europe, and AO Air to Australia respectively. Now its $2.05, $4.22, and $5.55 respectively. In spite of that, we have no intention of raising subscription rates.

TMQ advertising

If you are interested in reaching a dedicated TRS-80 audience, consider *THE MISOSYS QUARTERLY*. If you have a TRS-80 Model 3 or 4 related product to sell, you can reach these buyers by placing your advertisement in our publication. TMQ is read world-wide. Our subscribers are predominantly in the United States; however, we do have a significant number in Canada, Europe, and Australia. Space rates are as follows:

Full page	$250
Half page	$150
Quarter page	$100

We accept only black & white ads; however, ads for our inside covers are printed in the same color as the cover (TMQ alternates between PMS colors: green 354, purple 266, blue 293, and red 199). If you would like to place your ad in *THE MISOSYS QUARTERLY*, give me a call.

PD Software Librarian

Vic McClung has volunteered to be the librarian for the collection of TRS-80 public domain diskettes. Henceforth all requests and contributions be directed directly to him at:

Vic McClung
914 Crescent
Sikeston, MO 63801
USA

Out of print TMQ's available

For out of print issues, we are providing back issues of *THE MISOSYS QUARTERLY* via copier reprint. The price is **$12.50** plus $2.75 S&H in the U.S. and CANADA. For foreign zone D, the S&H rate is $5.50; zone E is $6.50. The price for regular back issues still in print is $10 + S&H. We are currently out of print on all issues except II.iii. Here's a synopsis of past issues:

Volume I See the index in this issue.

II.i David Hall on the 64180; Gary Phillips on XLR8 & 4P; Doug Tittle on sorting PRO-WAM data; WORD with DW II.

II.ii Extended DATE$ of Model I LDOS 5.1.4; Input SUBroutine for QuickBASIC; BASIC Interface to @EXMEM; HIRES Graphics for MC; Focus on speed.

II.iii Headline driver for I/III LDOS; XLR8er installation; CTL255 filter for PRO-WAM; FIXBANKS for your XLR8er; 4P boot ROM disassembled; C bit fields.

II.iv Tapes, Disks, and CMD files; Searching with DSM4; EXMEM<>BASIC interfacing; Testing printer ready; Revision to HIBANKS; WAMDUMP: convert a PRO-WAM /APP file to /CMD file; SYSDRV: A system drive other than :0.

DISK NOTES 3.1

Each issue of *THE MISOSYS QUARTERLY* contains program listings, patch listings, and other references to files we have placed onto a disk. DISK NOTES 3.1 corresponds to this issue of TMQ. If you want to obtain all of the patches and all of the listings, you may conveniently purchase a copy.

DISK NOTES is priced at $10 **Plus S&H**. The S&H charges are $2 for US, Canada, and Mexico, $3 elsewhere. If you purchase DISK NOTES 3.1 with the coupon which accompanies this TMQ issue, you can save $2.50; the cost then being only $7.50 + S&H.

Fixes in this issue

I did away with *The Patch Corner* last issue, because the number of patches had dwindled to a handful. This issue again has only a handful, but here's a reference to them.

BC54/FIX - For EnhComp's BC/CMD
BC64/FIX - For PRO-EnhComp's BC/CMD
LDOS53/FIX - For NEWCLOCK80 for LDOS
RD51/FIX - For XLR8er's Mod 3 RAMDISK/CMD
WAM23/FIX - For PRO-WAM
XLR8S12/FIX - For SYS12/SYS (6.3)
XLR8S1/FIX - For SYS1/SYS (6.3)
XLR8BU/FIX - For BACKUP/CMD (6.3)

TMQ Index

Elmar Von Muralt came through with an index to TMQ back issues. Oddly, Elmar used the exact tool I would have done for the job - our LB Data Manager. The index to TMQ Volume I is included in this issue. The format is topics sorted alphabetically, "title" sorted with topic, issue, and page number. I want to acknowledge Elmar's efforts and thank him for that needed input.

Market Research

Finally, here's some feedback on our external hard drive market research request. We were swamped with requests for this package. The level of interest was far in excess of what we had expected it to be. What may have helped is the query went out to over 11,000 folks. The response level was 1-2%; some of those looking for more than one package. Obviously, our proposed package was considered acceptable and I now have an expert individual exploring the design and engineering of the host adaptor, our own Joe Kyle-DiPietropaolo. Other than this, I have no news of a proposed availability; but anyone providing written or aural feedback will get a mailed announcement in addition to any posting in TMQ.

To recap the proposed hard drive package, here are the tentative specifications: 20 megabyte half height drive (Seagate ST225), PC-type hard disk controller (Western Digital WDXT-GEN) with on-board MS-DOS driver & formatter, MISOSYS designed TRS-80 host adaptor, Leadman drive case for two half height drives e/w 60-watt power supply (switchable 115V/230V) and fan, device drivers for both LDOS (model III mode) and LS-DOS (model 4 mode) along with archive/restore utility, and our diskDISK software. Our target price is still $495.00. We are also planning a hardware clock on the host adaptor as a low-cost option. The beauty of this configuration is that everything except the host adaptor is re-usable in an MS-DOS PC environment. Thus, if you can your TRS-80, you have a 20 Megabyte drive with controller totally reusable in a PC.

MISOSYS is ready to ship the off-the-shelf components now to those folks looking to acquire a hard drive, controller, or external case. Perhaps you just want to add a 20 Megabyte drive to your PC. Or maybe you are a hacker wanting to

upgrade your Radio Shack 5 Meg drive to a 20 Meg drive. Check out the next section of *The Blurb*.

New Product Announcements

PRO-WAM Programmers' Toolkit

We are now shipping the toolkit for programming under the PRO-WAM environment. The package includes revised documentation which fits into your new PRO-WAM Release 2 binder, a revised WINLINK device driver which supports programmed invocation of applications, a WINDOW/CCC function library revised for MC, and an assembler source code file for the PHRASE application to better illustrate how to write a PRO-WAM application. Order M-51-225 at $29.95 + $2S&H (US).

Mister ED Revised

I have made a minor change to the TED module of Mister ED. This fixes up the screen buffer handling problem that was corrected in LS-DOS 6.3's TED/CMD about 18 months ago. This required re-assembling the file as it couldn't be patched in. I had to work hard to compress the TED/APP code to allow room for the code revision. I'll make this a free disk refresh, but please include a return address label when you send in your disk.

I have also revised the Mister ED documentation to 5.5" x 8.5" format. If you have a PRO-WAM Release 2 half-size binder and want to get a Mister ED manual which fits into it, return the Table of Contents page of your existing full-size Mister ED manual with $5 and I'll send you the new manual.

Hard Disk Drive Components

Because of our pending development of an external hard drive package, MISOSYS is now stocking components for its assembly. We want to make these available separately. Here's what we have along with your cost:

1. Seagate ST-225 hard drive, 20 Megabyte formatted,
   $225 + (5 lbs) $7.50/$10.50 S&H
2. Western Digital WD-XTGEN drive controller,
   $75 + (1 lb) $4.00/$6.00 S&H
3. External hard drive case e/w 60 Wt ps; for 1 full or 2 hf ht,
   $125 + (10 lbs) $13.00/$18.00 S&H
4. Items 1&2 plus connecting cable set (data & control)
   $285 + (6 lbs) $8.50/$11.50 S&H

Note that S&H figures are US 48 States/Hawaii&Alaska. Outside US please write for shipping charges.

Standby/UPS Power Supply

MISOSYS has switched to operating three of our systems over to a standby power system. Summertime, in our area, is fraught with thunderstorms which result in temporary power outages, not to mention surges, brownouts, and other nuisances of clean power. I finally got tired of having to shut down when the skies darkened for fear of getting caught with an unexpected power outage. We just couldn't operate any longer. If your area is subject to unexpected outages, or has problems with clean AC, I suggest you consider a standby system or an UPS. The difference between a Standby Power System (SPS) and an Uninterruptable Power System (UPS) is that an UPS is always feeding your system with regulated, battery driven power whereas the SPS "rides the line" and switches over to the battery backup within moments of a brownout or blackout.

This issue includes data on the line of UPS' and SPS' we are offering. Unless you are hyper-critical, I don't see a need for an UPS over the SPS. The UPS is inherently more costly. The PTI Turbo/2 line of SPS' we're carrying switch to backup under 1 millisecond and also have surge protection. Most, if not all, computer power supplies should be able to hold their DC voltages for much longer than a millisecond when the power is cut. So a SPS is sufficient for most folks, and economical to boot. Our pricing on the two SPS packages recommended by us are:

PTI Datashield PC200A	$299 + S&H
PTI Datashield Turbo/2 300A	$469 + S&H
PTI Datashield Turbo/2 450A	$499 + S&H
PTI Datashield Turbo/2 625A	$549 + S&H

These weigh about 37 pounds each so shipping will be by UPS ground only, and only in the 48 United States. Shipping will vary between $5 and $18. The PC200A unit is recommended for floppy based systems; the Turbo/2 450A will handle a bigger 286 or 386 class machine. I have one 450A unit feeding Brenda's AST Premium/286 (40Meg drive) and Radio Shack Line Printer V. The PTI Datashield SS400 and SS700 systems are true UPS'; but prices are considerably higher ($750 and $975 respectively).

Family Update
by Brenda

Roy has asked me to write the family update again. He is so busy trying to get the rest of TMQ ready to go to the printers before we go away on vacation. We will heading down to Miami for a few days. Since it is getting too expensive to fly we will be driving. On the way we will be visiting my family. We will stop in Gainesville, Florida to see my brother and his family. While we are there my parents will be taking three grandchildren (our two girls and my brothers' boy) to Disney World. We will then be on to Lake Placid to see my grandparents and show them the new baby. After a short visit there it will be on to Miami for a couple of day before it is time to head home.

Speaking of the baby, Roy has given me the pleasure of announcing the arrival of our son, Benjamin Charles on June 9, 1988 at 6:44 p.m. He weighed in at 7 lbs. 12 oz. The girls are very happy with their new brother. In fact, they got to hear him cry over the phone when he was only about 30 minutes old.

Benjamin was born the day after Stacey's 5th birthday. When it was getting close to June 8, I was just hoping that he wouldn't decide to make his appearance on her birthday. Things couldn't have gone more smoothly. I went to pick my mother up at the airport about 11:30 a.m. Later that afternoon we went to the hospital. Within and hour and a half after arriving, Benjamin was born. We got the birthing room (it has a more homey atmosphere) and the nurse midwife that I had seen through all three of my pregnancies got there to do the honors. Roy even got to take a more active part by cutting the cord.

The next day when Roy brought Stacey and Stefanie to the hospital, the first thing they did was jump up on the bed to see Benjamin. Stefanie about grabbed him out of my hands. They spent some time with us, then Roy took my mother and the girls down to Georgetown for some lunch and a little shopping trip. We went home on Saturday. One of the neighbors had made up a banner saying "Welcome to the world, Benjamin" and another baked a cake. Stacey and Stefanie couldn't wait to show off their new brother.

We have now tried to get more into a routine. We will be revising the routine come September. Stacey will be starting kindergarten, which means she will be riding the "big yellow bus" with some of the neighbor kids. Stefanie will be in her second year of pre-school. I will be working on a car pooling arrangement with some others. During my free time here, I hope that Benjamin will cooperate so I can spend as much of that time in the office to get my work done. That way I will have most of the afternoon to spend with the kids. I want to get Stacey enrolled in a dance class and Stefanie in a gymnastics class.

The timing on the mailing of the "Required Reading" flyers was a little off. A lot of the response from that mailing hit us just about the time I went on "maternity leave". Roy did his best to hold down the fort. In fact, the morning of June 9 I spent most of the time in the office getting orders written up.

All of us are looking forward to our well deserved one week vacation. One of these years, we'll have to take a longer break. But next week, we "rest" up, for when we return, this issue of TMQ will be back ready to be labeled, posted, sealed, bundled, and trekked to the post office.

Have you heard the news?

Brenda and Roy Soltoff are proud to announce the birth of their third child,
Benjamin Charles,
on Thursday, June 9th, 1988 at 6:44 p.m.
Benjamin weighed seven pounds twelve ounces at birth and was twenty one inches long. Stacey Elizabeth and Stefanie Diane welcome their new brother into the family.

Letters to the Editor

PRO-WAM, TMQ delivery, et al

Fm Clifton Duval: Just a note to say Thank You for your programming efforts and support of my computers.

I am enjoying PRO-WAM and Mister ED. I have a Model I and a Model 4. PRO-WAM is replacing many of the uses of my Mod I, which I bought in 1979 shortly after retirement. It is still satisfying to use.

TMQ Vol II.iii arrived here [Star Lake, NY] April 6th. If it arrives on the West Coast accordingly, the subscribers there should get it by next Christmas. My next subscription will be first class just to find out what difference it will make. [Note: TMQ II.iii was mailed at Sterling, VA on 2/29/88 -ed]

I would like to know how to choose between EDAS and MRAS. I'm not yet an assembly language programmer. Mostly need it for typing magazine listing and learning.

I am also interested in knowing how few of us there were who were interested in a Model I LDOS 5.3.

Perhaps these questions would be of enough interest to address in TMQ.

Fm MISOSYS, Inc: As far as the Model I update requests, we probably had less than 100 when we decided to stop taking inquiries.

In the case of beginners, I think that EDAS would be easier to utilize than MRAS. Most folks don't need to deal with relocatable segments. That development concept is useful when a programmer is either developing many programs which can re-use subroutines, or when many programmers are involved in one complex program package. However, if there is an intent to get into the C language with our C compiler, I would steer someone towards MRAS since that assembler (or M80) is needed for C with our compiler.

Market Research Feedback

Fm Dave Krebs: I received your co-op flyer yesterday and thought I'd put in my two cents on your external HD project, so here goes ...

First, I can understand your desire to create a market for your fine HD drivers and diskDISK, but is it too ambitious a project, too late? Let me explain my situation ... I have a R/S 5 MB HD with your RSHARD package, and your diskDISK (mod4 only). I would like a bit more storage and although 10 MB might be enough, I've been looking at the adds for the Seagate ST225's with controller that have been in the magazines with controllers (MS-DOS) for $250-300. I look at these and wonder - if I bought one of these, could I switch the bubble for my 5 meg, have my increased capacity and transfer it to an MS-DOS machine when I get one. Notice I didn't say IF, but WHEN! I almost wish my old (1983 vintage) M4 would give up now, I might sweep up one of the 4D's selling for 500 bucks. I certainly don't want to think of replacing my software library with MS-DOS stuff! My model 4 does about all I want even though it may not have all the bells and whistles of the new '286' machines (I'm doing this at work on a 70 meg Tandy 4000 with WordPerfect and a laser printer).

So, that's the Model 4 users dilemma! How much do I keep plowing into hardware and software! Do you perceive a market for new HD users, I don't. I see your market as the diehard users who will use their hardware 'till it croaks, but need more storage for the HD they already have. Don't get me wrong, If I could dump my 5 meg unit for, say, $200, I'd line up to give you $500 for your new 20 meg, especially if I could use it in a DOS machine, with new controller, of course, later. As you can see, I place a limit to the value of increased storage. Your unit sounds great ... I like the small footprint. Would a conversion bubble for the R/S cases be a viable product? Probably not, it sounds like a hardware hackers project!

I hope this rambling has given you the feedback you're looking for. Thanks for your continued support. I'm always looking forward to the next issue of TMQ.

By the way, I have a draft of a little piece you may want to consider for TMQ - it's a speedup for the OLD model 4's. I ran across it a couple of years ago and finally had the guts to try it when I had the case off installing SmartWatch. It has to do with bending up pin 7 of U3. I got a 20% increase in a integer loop counter. I'll send it along on a disk in a week or so.

Fm MISOSYS, Inc: Dave, From the feedback I have received from that mailing so far (and its only the 3rd of June), I will have plenty of orders to justify the hard drive

package. The Model 4/III is far from dead. A lot of folks have a very large investment in software which meets their needs. The machine works for them. If they can forecast a need for a reasonably priced hard drive, then it keeps them operating better for longer with no additional investment in software. Some folks never consider the money they have invested in software and the costs to acquire replacements on another operating environment.

Now I recognize the concern with re-using the investment over in the MS-DOS world. That's why my fundamental approach is to use everything from the MS-DOS world to build this package. Not only would the drive be re-usable, but the controller as well! We're going to use a controller which can plug into an IBM PC or clone. Thus, when you replace your 4 (or add to it), you can take the drive, controller, and connecting cables out of the external case and plug them right in to a PC clone. The software for the MS-DOS world is in a ROM BIOS which is part of the controller. How's that for protecting your investment.

On the other hand, in your case you would just want to replace your 5-meg bubble with a 20 megger. The LDOS forum (Compuserve PCS-49) has had some info on what needs to be done there.

It's a tough question to consider just when you stop adding to your present machine and replace it with a "newer" model. The same question arises when you ponder whether to trade in your old car for a newer one or just fix up the current one. When the water pump goes do you say let's fix that and get rid of it, or do you say I'll continue to fix it because it served me well. You must also look at the long term. I personally would not want to design a dead-end peripheral. Any printer you purchase for your 4 should be usable under MS-DOS. I feel the same way about a hard drive.

Incidentally, we can take either Model 4 disks or MS-DOS disks (360K, 1.2Meg, or 720K 3.5"). The important point is to label the disk as to its configuration.

Fm Laird G. Jackson, M.D: I recently got back into the TRS-80 support circle by being forced to install my nearly one year old copy of LS-DOS which extends the operating system calendar through 1999. At that time I had to call LSI about LS-DOS 6.3 and got a look at the MISOSYS information. This eventually led me to your announcement about 20 megabyte disc drives. That device sounds interesting at the target price that you're guessing. We have at least a half a dozen Model IV's or IV P's sitting around doing allwrite, PFS file/report and other things. I did get a copy of Pro Wam and have had a quick look at that and see how it can be useful for us. I have sent in a subscription for the quarterly and I am interested in both a hard drive and perhaps even that so-called accelerator. I'd like to ask a couple of questions about the drive. Most of the Model IV's that we have have been converted so that they presently use half height floppy drives, two in at least the bottom holes. Why couldn't you install the hard half height drives in a similar fashion? Do these drives need to be connected through the RS-232 port? One of my somewhat nagging concerns is since I already use a rather clumsy and not all together reliable external drive, I'm wondering what I'm going to do with that data. I recently went through an exercise of transferring a large amount of fairly critical data off of one of the Bi-Tech's by pumping it out the printer port in the PFS print option in file through the parallel port and in through a parallel to serial converter and thence to a named file in the Unix system. If your drives could be set up internally it would make the cabinetry neater and allow easier salvage of existing data. In any event, what would using one of your 20 meg drives offer in terms of transferring information out of an older hard drive. Would there be some reasonably simple means of transferring that information or would I have to use some Rube Goldberg arrangement or a program such as the Powersoft version of backup/restore which creates hard disc mirror images on a series of floppies and then puts them back on another hard disc. I'm sorry to trouble you with these mundane questions but not being much of a computer expert, I get nervous when I have to transfer data through methods that are unknown and untried to me. Thanks for whatever information you can give me. I am interested in your products.

Fm MISOSYS, Inc: Dr. Laird, There is no reason why a hard drive cannot be designed to operate internally within a Model 4 computer; folks have been doing that ever since the early Model III days. The question is whether our initial design should be internal; it should not. We have to assure that sufficient business will exist to justify our presence in the hard drive add-on market. In order to ensure our success at a reasonable price, we must first engineer the one package which would be in most demand. An internally mounted drive would require either (1) an additional power supply (or replacement of an existing supply), or (2) use of an extremely low-power drive which is much more costly. If we get our feet wet with the external drive and find sufficient demand exists to justify bulk purchase of raw materials for an internal addition, we would of course pursue that approach as well.

Hard drives are not connected through the RS-232 port whether internal or external. They are connected via the 50-pin external I/O bus.

The way to port data from one hard drive to another (since you can really have only one connected to the I/O bus at a time), would be to back up all files to floppies, then restore the floppy files to the new hard drive. The software we would be providing would include such an archive and restore facility.

Fm R H Hollenbeck: Just received your "Required Reading" publication. Great stuff!

I would like to offer a couple of ideas regarding your request for input about developing an external hard drive package for the Model III/4. Before I do I want you to know that I think of

MISOSYS, its officers and employees, as friends. I only wish that we had met earlier.

As an accountant, I must question the advisability of spending time and money to develop anything for the TRS-80. It is doubtful that you will be able to amortize the R & D costs under the new tax laws.

No doubt you have definitive information regarding the number of TRS-80's that are out there. But, do you have any feel for how many owners of those machines either already have a hard drive, or have one full-boat on floppys and are therefor probably not interested in acquiring a hard drive? No doubt this is the reason for your request for input. (I am running a 128K, 5meg Model 4, and probably wouldn't be interested in acquiring another hard drive.)

If most of the TRS-80 owners are like me, I will keep using my Model 4 until it gives up the ghost - but, I can't justify spending significant amounts of money to up-grade the system. I will continue to maintain and repair it; I may even add an inexpensive (fun-type) peripheral or two. I think that any major expenditure (and $500 is major to me) can be postponed, and the money saved and put towards an MS-DOS machine when the time comes.

Back to the old tight pockets accountant's wet blanket. I would recommend that your break even analysis be reviewed, and if you can't crack the nut with one half of projected sales, leave it!

I realize that developing a new hard drive would probably be a lot of fun. But friends, can you make a buck doing it? I don't think so. I think that you can use your resources more profitable in making the transition to support of the de-facto standard.

I will continue to subscribe to the TMQ as long as you publish it, and make an occasional purchase (I've got most of your stuff). None of the foregoing is intended to detract from what you folks have done; it's just that I don't want to see good people get hurt!

Fm MISOSYS, Inc: I appreciate your concern; it's great to have you folks looking out for my interests. Actually, I really wasn't too sure how receptive my hard drive idea would be; that's exactly the reason for the inquiry. I am overwhelmed by the response to my market survey. And that only went out to about 12,000 folks. I have more than an ample response; some even willing to put their money down now! Thus, I am going forward with the engineering.

To tell you the truth, there really isn't much to deal with. Most of the components of a nard drive package are off the shelf components. A package consists of case & power supply, drive, controller, host adaptor, software, and cabling. Out of the five components, the first three are standard. The host adaptor is a minimal part of the overall package; its the only component which needs designing. As I have already done a few software drivers which "talk to" the type of controller I'll be using, I only have to make small modifications to my existing software package. In order to manufacture connecting cables for the XLR8er board, I have already acquired a small cable maker device. Replaceable jigs allow it to support various kinds of Insulation Displacement Connectors (IDC) used in cable making. The only cable I need to manufacture is the cable which connects the 50-pin I/O edge card connector on the Model III/4 to the hard drive.

So it's not a lot of work. In fact, the entry into this area of hardware has other advantages. The Seagate ST-225 hard drive is one of the most popular hard drives around. We expect to also offer the drive by itself, or as a hard drive upgrade to a floppy-based PC. Since we have to buy the drives in quantity to support our OEM package, it makes sense to broaden the base of potential sales by offering the drive separately to those who want to purchase a separate drive. I also recognize the fact that the major portion of MISOSYS' revenue still comes from the TRS-80 market. By adding a reasonably inexpensive hard drive package to that product line, I also add an incremental amount of revenue and profit; this makes us healthier as a company and extends the time which we can stay in this business.

Fm Gary Lee Phillips: Roy, Just received your sale flyer full of interesting news. Can't say I'll miss LSI as long as you'll be supporting 6.3, but I am surprised that they gave up quite this soon. Does that mean you will now be in control of "piracy protection" policy and site licensing Policy, etc.?

All model 4 users owe you a tremendous debt of gratitude, and I hope they will continue to support you as much as they can. Just making the effort to pull this flyer together for us is a fine example of your continuing contributions. It was interesting to see who was included and who was not: have Powersoft and Aerocomp now abandoned us officially?

With respect to your hard disk project: YES, I want at least one if you can come in near that $495 price estimate. If it works with a Model 3 and LDOS 5.3, I might even he able to buy two, but in any case I can promise to take one of what I'm sure will be the best-documented and most solidly-designed hard disk for the TRS-80. If possible, please include the write protect switch in your design: that was one of the best features of the Radio Shack hard disks.

It would appear that Tandy is making model 4 Fortran hard or impossible to get, at least from the remarks on the LDOS Forum. I don't suppose they would consider transferring their distribution license to you, but it might be worth asking--sales of RATFOR could pick up if you had the Fortran available as well.

Fm MISOSYS, Inc: Yes, Gary, MISOSYS is now in control of site licensing. I have been much too busy to evaluate any change in policy. I am contemplating a "Personal

Site License" for those folks who have a small number of machines. I would suspect that the next TMQ would have more on that thinking. Also note that we don't yet "own" LS-DOS 6.3; our agreement with LSI only gives us the right to sell and support the product, although Bill has given me leeway to make any changes I deem necessary. Of course 6.3 is just one product of ours; it doesn't demand immediate attention. I am still trying to clean up all of the little things around here so I can concentrate on developing a new release of our LB Data Manager.

Powersoft and Aerocomp were apprised of our flyer. I never heard from Dennis Brent, although since he had recently sent out a flyer to his customer list, I really didn't think he cared to get into another one. I also assumed from the content of his flyer that Powersoft was shifting to a very mild exposure in the TRS-80 community. On the other hand, I spoke to John Lancione a few times. He was contemplating taking out a few pages of ads in our flyer. Nothing ever gelled. But I don't think that they are leaving the TRS-80 market.

As for Tandy's FORTRAN, I wouldn't even begin to approach Tandy about transferring their Microsoft FORTRAN license to us. If you have ever worked with Tandy, you would understand how futile that would be. But various folks at Tandy do get TMQ; perhaps they will sit up and take notice.

Fm Mike Hollenbeck: I am very interested in your little bit of market research as appeared in TMQ II.iv, page 5. I had toyed with the idea of building a "host interface" as you called it for some time. I saw no reason that the IBM type controllers would not work with just a little ingenuity.

I would like to see you offer other versions of this product. Have you considered offering an internal kit? This would help with your problem of getting FCC certification for the external drive. Another thought is to offer the parts such as the host adapter, drive and controller or enclosure a-la-cart? As a hardware hacker, I have collected parts here and there and at this time could make use of the host adapter only (and a little software).

Please take these ideas into consideration when you plan to market the product. Good luck!

Fm MISOSYS, Inc: Mike, I do expect to make available individual components of our hard drive package. In fact, you will see some of those prices in this issue. I also have plans to think about an internal drive after our external package is available and purchases are underway. Just give me time! I also may make the entire package available as a kit of parts. Guess what that would be for?

Fm Marc A. Williams: Roy, I got your consolidated advertising mailer today, and I just wanted to give you some feedback on it and on your hard disk marketing plans.

First off, congratulations on the smashing idea of the mailer! I have been pulling together old 80 Micros (I guess they're ALL old now, tee hee) to see if there are any last toys I want before the vendors turn to vapor. Your mailer provided some nice reassurance in that regard.

I currently own a Radio Shack 5MB hard drive, to which I have added a 20MB secondary built up from surplus parts (drive, case, cables & power supply). Nonetheless, I would still be interested in a $495 20MB system plus software. Although I don't know of any at 20MB, I would suggest you stay away from RLL controllers, however. Friends of mine who have used them on pcs have often had data reliability problems -- probably because the RLL system was trying too hard to "squeeze" storage from the platters (20MB drive pushed to 30MB, etc.).

As for other products, I would absolutely die for a reasonably priced archival backup system. I don't know if it's just me or what, but I find my floppies tend to be difficult to read after they get to be 5-6 years old. Refreshing them every few years (I have several hundred) is a bit of a chore. I don't know what archival method would be the best over long storage times: video tape backup, dedicated tape drive, removable hard disk, or some variant on floppies with perhaps more redundancy, etc. I've seen quite a few offerings for PCs, and some that even purport to hook up to a floppy disk controller (only the software to write for TRS-80 use!). My primary concern is long term storage. I have two TRS-80s (a M4 and a MIII) as well as a third, older machine. Each has its use, or maybe I just don't like throwing things away (I certainly can't sell them for anything!). Each has a sizable investment in software which I don' t want to lose.

Anyway, keep up the splendid work!

P.S. I 'm still trying to get a Seatronics speedup kit. It turns out Sylvester Technologies does not now, nor ever has been a distributor for them -- just a satisfied customer.

Fm MISOSYS, Inc: I was very pleased with the response of the mailer too, Marc. We expect to engineer another one in the fall.

I stay away from RLL controllers in the MS-DOS world as well. I don't think one would work in the TRS-80, anyway. The problems which some MS-DOS folks have experienced when using an RLL controller is they have tried to use just any drive. That won't be too successful. The RLL encoding requires a greater degree of precision on the hard drive platter, heads, and electronics and a higher degree of quality of the plated material making up the magnetic surface. Just any old drive won't cut it. A drive must be designed to work with an RLL controller. A stock ST-225 won't cut it. But many folks have tried to use that drive with an RLL controller then complain about losing data.

I do expect to explore some means of data archiving on other than floppies. But that will not take place right away. I think that there may be two solutions. One is a FloppyTape device which works off of a regular floppy controller. That would require access to the external floppy connector. One would have to be added for a 4P. The other is a removable half height hard drive as the second drive of the package. After I get the hard drive package out, I'll be in a better position to investigate these two facilities.

Fm Charles A. Ainsworth: Roy, You ask for suggestions regarding other products readers would like you to offer. Here are some ideas. Note that I neither ask or expect a personal reply to this letter which I am writing in the spirit of helpfulness for you to pool my ideas with those you may get from others.

One idea is the possibility of offering a ribbon reinker. Admittedly, its a product with a relatively low price in proportion to weight and volume but, if shown around a bit, might generate some sales. I have one I got from Computer Friends, 14250 NW Science Park Dr., Portland, Oregon 97229, (503)626-2291. They had an ad in 80 Micro of June 1988, p65.

About a year and a half ago I purchased a couple of Tandy DMP430 dot matrix printers; nice job, but the price of the ribbons! A whole new ribbon costs $15.95 and the carton says it's good for 4 million characters. Well... maybe. Anyhow, I seemed to be changing ribbons quite often and it really irked me to throw away all that chunk of hardware when all the good I'd got from it was a thimbleful of ink. Then I discovered that Tandy has refills at $15.95 for 3 so I tried those. But they involve opening up the cartridge and messing around with inked ribbons and getting oneself smeared with ink despite the plastic glove that comes with the refill. Besides, there again, I still didn't feel a thimbleful of ink was worth over five dollars.

Finally I got one of Computer Friends reinkers. Beautiful! Now I can reink a ribbon for, at a guess, five cents or so. The device works well, but perhaps a bit slowly, but that doesn't matter as I start it inking and go off to do something else. A single run through a ribbon takes 21 minutes and I have found I prefer to run it through twice, total of 42 minutes for a reinking. The number of reinkings a ribbon can take before it starts to wear is as yet an unanswered question. I have ribbons I have reinked some ten times and I can't detect any signs of wear. Naturally, as soon as wear shows, I'll have to use a refill.

One of the points that I like is the system of applying the ink. Some reinkers, in the past, had a revolving sponge roller, moistened with ink, bearing against the ribbon as it went by. I understand that this sponge roller arrangement was messy. Now, my inker doesn't use a roller, but the ribbon runs tangent to a small vertical cylindrical ink container with a minute hole in it through which the ink exits onto the ribbon. Quite neat.

I seldom or never hear of other computer users reinking their ribbons, but I feel sure that if the product were shown around a bit many might be interested. Maybe Computer Friends' advertising has been a bit too low key towards the back of 80 Micro. Now, if you could get folks talking about it in TMQ as they are talking of the XLR8er, maybe interest would be awakened.

Another item which may interest you: A power strip with a cord to connect to an outlet, the strip provided with several outlets for plugging CPU, drives, printer and other peripherals; with a main switch on the power strip to connect and disconnect CPU and peripherals simultaneously, and preferably with good surge/spike protection. I have been using such a thing for several years in an area where a thunderstorm sometimes produces enough fireworks to set off my telephone answerer or cut off the power supply, with no problems whatsoever from surges or spikes. Compare with a friend of mine who operates an office nearby and has a PC, hard drive, laser printer, etc. without any surge protection. A thunderstorm discharge blew the electronics of his expensive laser printer and it seems will cost him around $500 to repair.

Turning to another aspect: Your piece in TMQ says you have dealer arrangements with a number of distributors so if a buyer isn't in a hurry, you may be able to obtain a desired component at a reasonable price. That sounds interesting, and I can well believe that when you talk of a reasonable price, you really mean it. Also, if I need something and you can make a buck in the process, I'll be glad to cooperate.

However, the question immediately pops up in my mind: Just what sort of items are you referring to? There is a bewildering array of hardware out there, all the way from complete computers and systems through small things like chips, listed in hundreds of pages of advertising in computer magazines and in catalogs. Perhaps some orientation for your readers may be in order; otherwise I visualize your being deluged with numerous inquiries about items you cannot supply.

Fm MISOSYS, Inc: Ribbon re-inkers work well. I have two of them. But I don't think Computer Friends is low key on advertising. I see a card on them in practically every card deck I get. There's also about two other outfits which make re-inkers. Tandy sells a power strip. So does practically every computer store. I have access to them. If more folks want that item, let me know. I can start carrying power strips.

Now to give a list of the kinds of products we have access to would take most of this issue of TMQ. But you're right, unless I make some kind of statement, no one will have any idea what I'm talking about. We are starting to carry an Uninterruptable Power Supply (UPS) because I think that's a definite item to have. As far as other things, let me provide a sample.

We can supply Western Digital controllers; Seagate and Priam hard drives; modems from Novation, Hayes, and Anchor Automation; Intel processors and board products; Nec monitors; Mountain tape drives, Boca Research board products; American Power Conversion UPS; AST computers and board products; Acer, Televideo, Wyse, and Toshiba computers; Products from Alps, Citizen, Dataproducts, Diconix, Emerald Systems, Magnavox, MiniScribe, Novell, Panasonic, Paradise, Proteon, Samsung, Star Micronics, U.S. Robotics; also software from biggies such as Aldus (Pagemaker), Ashton-Tate (dBASE,etc), Bedford (Accounting), Borland (Turbo-everything), Brown Bag, Computer Associates, Fox & Geller, Funk, Generic Software, IMSI, Infocom, Lattice, Logitech, Lotus, Microsoft, Micropro,Norton, Paperback, Phoenix, Polygon, Quarterdeck, Samna, Softlogic, Software Publishing, Wordperfect, and Zsoft. This is a sampling.

Fm R. M. Doerr: Gentlemen, Thank you for the recent announcement folder.

My DOS 6.3 CS # is 17888; the level iS L+. I have acquired another Mod 4; the DOS 6.3 CS # for it is 17978; it is at level J. May I send both master disks to have them upgraded to the latest level, and keyed alike? What is the latest level? Is it significantly ahead of Level L+?

You ask what software readers would like for the Mod 4. I would like:

(1) Comprehensive hardware diagnostic software.

(2) A powerful printer controller that operates from DOS and could be used as SYS13 - to set up the Epson FX-80 printer. I have one that does the job, but only for a few settings, as it is designed for the MX-80. It does the main job, setting for condensed type.

(3) A good beginner's book on C. After having help to learn FORTRAN in 1962, I taught myself PL/I and BASIC, so I think I could teach myself C.

(4) Upgrades for Scripsit Pro a) to enable executing a wide range of LIB commands from within it; b) to formfeed, rather than do a sequence of linefeeds, at the end of each page; and c) means to convert Scripsit Pro documents to ASCII for communication in this world of MSDOS.

(5) Means to print a graphics file from BASIC (and GBASIC), for example, to insert a line that is set in a fancy (DOTWRITER) font, or to install a logo at the head of a page.

(6) an ASCII editor that would allow 132-character lines, DOS commands, condensed printing, and a full range of ASCII characters. (David Goben's enhancements to Disk Scripsit may enable it to do these things.)

(7) An upgrade to enable big Multiplan sheets, using 128 K.

I would be interested in a reliable 20-Meg hard drive for the Mod 4. I think $595 would be OK for a ready-to-plug-in-and-run external drive, including power supply, controller board, cable, cabinet, good, fast Seagate drive (preferably self-parking), battery-backed-up real-time clock, practical software to boot (date and time ready) from the hard drive, good, dependable PDS software to circumvent the 8 drive by 256 files-per-drive limitation, and any DOS enhancements needed for the hard drive.

Fm MISOSYS, Inc: Wow, what a list. Let me put that into perspective. Most of those items are beyond the resources of MISOSYS. Perhaps some of our readers can render assistance. Item (2) should be easily satisfied by one of the public domain PRO-WAM applications you can find on our Compuserve forum. MicroLabs may be able to shed some light on (5) since they manufacture a graphics board. As far as (7), I believe that Multiplan already makes use of the extra 64K to allow about a 90K spreadsheet. As far as the hard drive, our plan was for everything but the clock at $495. No DOS enhancements are needed; we will be bundling our diskDISK product with the package.

Fm Jay Carman: I think it is a very good deal for customers and you, that you now are the seller and servicer of LS-DOS 6.3. Congratulations on this transaction.

I would be very interested in the hard drive you mentioned in Required Reading. My used Megadisk Just conked out, and I don't want to get another Aerocomp hard drive if I can help it. What would be the possibilities of mounting one or two internally in a Model 4?

Disk Disk still doesn't work very well with my Aerocomp 30 Meg drive even with the patches. I really didn't expect it to, since I think something else is also messed up with the controller or software. I have been unable to format the drive any configuration other than the one it came with - 3 drives of 10 meg each. When it was shipped to me the config/sys was in the hard drive and not on the disks sent to me. The answer I got on the telephone was that they were not in the business of giving hard disk tutorials. I managed to get in to the hard disk by studying RS and Powersoft hard drive manuals and the sheet of paper sent by Aerocomp. I filled all free space with Disk Disks, located the good ones on the higher cylinders, left the good ones packed by bad ones, and deleted those on the lower cylinders. I managed to get about 8 good 180k Disk Disk on each of the three partitions, and have access to about 2/3 of the drive. It has worked fine up to this time.

I have not had a chance to hook up XLR8er on my office Model 4 yet, but I will let you know how it does, and the motherboard version as soon as I do.

I will be able to transfer some of my Profile 4+ programs over to Little Brother, but some I won't be able to at this time since I use the data in some basic programs and don't know the fine art of blocking and deblocking as required by Little Brother

data files. The program appears to be a very good one, and I think I will enjoy using it. I have to delete some of my boot up utilities to use it, but I knew that before I purchased it.

The more I use Prowam the more I appreciate the program and its possibilities. I boot up with it as part of my configuration. Of course when I use Profile 4+, I have to reboot afterwards.

I am still recovering from my total hip replacement, and haven't kept up with my computer dabbling too well, but I might as well get started with another phase. Please send me EhnComp M-20-072 for the Model 3. My check and notice are enclosed.

Fm MISOSYS, Inc: I suppose you could add two half height hard drives internal to a Model 4 if you add another power supply and a fan. We'll start on an external, first. You may find Daniel Srebnick's article in this issue of some help in addressing a BASIC access of an LB database. Finally, if you remove PRO-WAM before you invoke PROFILE, you probably wouldn't have to re-BOOT.

Fm Harry G. Maurer: Roy, In response to your query on whether or not I would buy an external Hard Disk the answer is YES if the price range does not exceed the target price of $495.00. A price of $500.00 or over would be to much for me and I would not buy it. There is one other factor that would enter in my decision. The unit should be designed so that it can be a direct replacement for an existing floppy drive.

On other products: One suggested item is a reliable and economical external MODEM that supports the Hayes modem command set. As an example; a 300 to 1200 baud modem in the price range of $100.00 to $175.00; a 300 to 2400 baud modem in the price range of $150.00 to $250.00; and for the luxury minded customer a 1200 to 4800 modem in the price range of $275.00 to $400.00.

Just for your own information there is a 300 to 2400 baud modem called EMEX that is used by a few bulletin board systems and is sold for $225.00; however, the big sales pitch with this modem is that it is designed to be upgraded to 4800, 9600, or even 19200 baud by sending the modem in and getting it upgraded for a price around $100.00. The catch is that as of this date these upgraded boards are not available. If you are interested I can give you the name of the SYSOP and board number that you can contact for more info.

On the distaff side I would strongly recommend that you totally disregard any temptations to add printers to your business. The printer market is a real boondoggle and will cause nothing but grief.

Other ramblings from a weary soul: It would appear that MISOSYS is becoming the rallying point around which all TRS owners can get info. To support this group of TRS diehards would it be possible to have the 10 companies you mentioned in the recent release of REQUIRED READING supply you with patch data, new version release info and reported problems to be included in the QUARTERLY? I understand that to keep the noise level down for you it would be necessary to put a disclaimer in the section to contact the company supplying the info and not MISOSYS for clarification or help. To help support an effort like this I would assume that the supplier would pay for the insertion of his info and data. The only alternate would be to raise the subscription rates to $40.00 for first class mailings. Another thought would be to split the costs between a rate increase and the supplier of the data.

You are probably aware of this but I'll include it anyway. There is another totally dedicated TRS80 support publication called TRSTimes. Their first issue was put out in January 1988 and have published a March issue and a May issue. TRSTimes is published bi-monthly by TRSTimes publications, 20311 Sherman Way #221, Canoga Park, California 91306. (Maybe another company to add to the 10 you have already rounded up).

Just a few parting words about mailing the QUARTERLY. The postmark on the postage meter stamp was May 27, and I received, in my mailbox, the magazine on May 31. Needless to say I have nothing but praise for the timely arrival of the QUARTERLY. An interesting side comment - The quarterly was packaged in a very expensive looking and thick plastic wrapping and for some reason the postage meter stamp was placed inside the wrapping and as a result never got canceled. Also the cost was $1.85 for the mailing which indicates to me that your shipping and handling costs of $2.00 is very fair indeed.

Here's wishing you the best in carrying the TRS80 torch and also your family in the upcoming months.

Fm MISOSYS, Inc: TeleTrends carries modems for the TRS-80. Perhaps they have one to suit your needs. Also, my statement about being able to obtain other products did not imply that we would **stock** them; only that we would be able to order them for you. Yes, printer prices seem to change daily as new ones come on the market.

We are perfectly willing to accept various kinds of input to TMQ; make your suggestions to the other companies as I really don't have the time to ride honcho on that. We do the best we can rushing things here to make the deadline. I suppose I wouldn't be happy unless I was always two months behind on things.

Lastly, you misunderstand the concept of postage meter imprints; they need no further cancellation. We businesses pay for the privilege of using a postage meter. It saves us on the time to prepare mail, we always have the "correct" exact postage, and the avoidance of cancellation may help get the mail on its way a little sooner. Also, since the postal rates

went up, we now charge $2.75 for shipping a single issue of TMQ - still a bargain.

I use computers for...

Fm Henry S. Campell, M.D.: I own two model 4P TRS-80 computers with one DMP 430 printer and one DMP 130 printer. I use Scripsit Pro, Visicalc, and Profile 4 Plus. I know nothing about programming.

I use my computers for the following purposes:

1. As word processors since I write editorials for the Medical Society of Virginia.

2. To keep track of my personal finances.

3. To keep track of organizational lists such as the Democratic Party of Martinsville.

I am willing to invest $30 for a year's worth of THE MISOSYS QUARTERLY because I Know that You will Teach me a lot.

By the Way, where is Sterling?

Fm MISOSYS, Inc: Sterling is in Northern Virginia in the easternmost part of Loudoun County which abuts Fairfax County. We physically sit about 5 miles due north of Dulles airport where the Concorde makes two trips overhead each day. That places us along the Potomac river. We are about fifty minutes from downtown D.C. outside of rush hour. I wouldn't know how long it would take during rush hour because I wouldn't venture that way at that time.

Fm L. R. Boatman: I hope Brenda has had the baby by the time you get to read this letter. I also hope her ankle healed well. I've been out of town and just getting around to reading some of my backed up mail for the last three months.

Concerning your request for Market Research on a 20 Meg Hard Drive and priced as you stated in the Quarterly ($495.00) you have my interest and I will look forward to the availability of the system.

A few months ago I had a problem with the XLR8er and PROWAM. You hit the nail on the head with your reply. Seems that I failed to install the faster Ram (15Ons) in banks 1 and 2 of the 128K. I now load PROWAM into Bank 10 "PROWAM (BANK=10)" and have not encountered any further problems. The Ram is not available in 150ns in this area. Would you please tell me of a third party vendor who handles these chips and has directions for their installation in the Model IV, also the price if known?

The following is a short narrative on my use of the Model IV (I have two (1 with XLR8er and 1 with 128K, converted Model III). I use the machines during three months of the year to do Taxes (JAN-APR). The addition of the XLR8er on the DS Disk Drive Model IV and the use of MultiPlan reduced the amount of time the customer is serviced in half. I also do payrolls and book keeping for a number of customers. My children and I also do a lot of our school work (Learning is a never ending process). I keep a mailing list and copies of the clients taxes (1000+) and a Data Base. The use of PROWAM has been and is a valuable tool in the development of both.

The above information is sent in reference to the survey or Market Research concerning the HD for the Model IV. I don't know how much longer the Tax software people are going to continue to support the Model IV, but with your proposed package and with it being reusable in a PC environment, it has my interest.

Miscellaneous matters

Fm Ian Lange: Roy, Firstly, let me say that I have thoroughly enjoyed all of the TMQ's, from I.i on, as well as the Notes before that. While I don't do an awful lot of serious stuff with my Model 1 and Model 3, I do get a lot of useful information out of the journals, and I'm glad and relieved that SOMEONE still cares about us TRS80 users.

I trust all has gone well with Soltoff Jr. #3, which (I trust) will be in existence by the time this letter reaches you. I have always enjoyed those little personal bits, even when the baby throws up on you, or similar unfortunate accidents. Being a bachelor myself, I am very glad to keep these details at one remove with only nieces and nephews to worry about. Keep up all the good work! Pity the photos in TMQ II.iv didn't come out a bit better, but even from what I can see, they're obviously a pair of charming devils. Hope the third matches up the set.

I noticed in II.iv a note from Michael Rogers about bugs in Lescript 1.80 on a Model 1 - thank God! I had trouble with it too, and was starting to suspect a dud copy of the program, but then that's what I thought at first about an earlier version that I got, too (v1.68 or thereabouts). That one only worked on a Model 1, and crashed on a Model 3, whereas this one seems to be the other way around. Better send the disk back for repairs, I guess. Not that that is your concern, just thought I'd mention it in passing.

Well, there's nothing more I can think of for now, so, with my best wishes to all the family (Good Grief, they almost feel like part of my family!), Best regards to all.

Fm Lee C. Rice, Ph.D.: Friends, Thanks for TMQ II#4, which (as always) contains a wealth of information and help. In the past couple of issues you've mentioned that you want articles from readers. Would you be interested in one on

desktop publishing? MU Press publishes both a journal (**Philosophy & Theology**) and a series of volumes (**Mediaeval Texts in Translation**) which are produced on an HP+ driven by a Model 4. We spent a couple of years setting up the software configurations and writing our own utility software, as well as examining a lot of MSDOS text formatters (Word, TROFF, Word-Perfect, PC-Write, etc.) before ending up using the Model 4. I don't know if your readers would be interested in this sort of information or not. I am the author of several articles appearing in other micro journals, and would be happy to send you something IF you think readers would be interested.

I hope that Misosys will continue to produce software, since I'm running out of things to purchase from you. I already have purchased just about all you have for TRSDOS and LDOS (except the BASIC compiler, since I don't use BASIC), and just about everything for MSDOS as well. **Utilities** are probably the best prospect. MSDOS is a very kludgy DOS, and TRSDOS users come to it rather spoiled. Things that are easily done under TRSDOS and with the use of your utilities are often a virtual impossibility under MSDOS.

The blurb that you are working on a version of **Diskdisk** for MSDOS was good news indeed. MSDOS is profligate with space on hard drives. I've already had to move from 20meg to 40meg, and the end is nowhere in sight. The MSDOS path statements are clumsy, and subdirectories waste disk space, not to mention fragmenting the hard disk. Diskdisk86 would be a genuine blessing.

I'm also glad to see that you are getting adverts from other vendors which still provide support for the Model 4. The demise of **80 Micro** is no real loss - I suspect that **PC-View** will go next, since it cannot hold a candle to **PC World** (for general info) or **Dr. Dobbs Journal** (for programmers).

Remind your readers regularly that, when contacting a vendor who advertises in **TMQ**, they should always include the information that the advert was **seen in TMQ**. Many readers will forget if not reminded, and advertisers need and deserve that kind of feedback.

I have been enjoying your RATFOR, which I purchased both for TRSDOS and for MSDOS. Release 4.1 of MS-Fortran for MSDOS just arrived, and works splendidly with Ratfor, in addition to generating nicely optimized code. The RATFOR Manuals (introduction) listed a **User Group** in California which supports Ratfor as a language. I have written to them twice, both times sending postpaid envelopes; but I get no response at all (letters don't come back, so they are being delivered).

Any ideas on where interested users can go for additional RATFOR help and code? Our central VAX system is literally loaded with libraries of public Fortran subroutines (many of which I have converted to both MS-Fortran and Model 4 Fortran), and there is a RATFOR preprocessor for VAX Fortran also; BUT almost no subroutine libraries. I usually rely on subroutine libraries a great deal, since I never like to reinvent the wheel.

Fm Ken Strickler: This letter is to thank you and all the others who have supported the TRS-80 class (Z-80 based) of machines over the passed ten years. Even the notorious 80-MICRO covered the machine quite well until they went blind!

This list wouldn't be complete without mentioning LSI, POWERSOFT, PROSOFT, COMPUTRONICS, MICRO-SYSTEMS, just to name a few. The software contributed by these groups is mind-boggling if one just stops and considers the complexity. I wish to extend my thanks to Hardin Brothers, a STANDOUT PROGRAMMER, for all the wonderful short assembly programs he tutored us with in 80-MICRO.

Another special thanks to you, ROY, for the production of the MISOSYS QUARTERLY continuing the support of the Z-80 machines. During one of the "conversations" in TMQ, you mention, that some of your readers do not have a link to COMPUSERVE. I fall into that group, and would like to thank you for all the reprinting that you do. The contributions that I have gotten from the magazine more than pay for themselves, even without counting the FANTASTIC deals on software offered. I have bought all but a few of the items that you have offered, including LDOS 5.1.4 for my model I's, LDOS 5.3.0 for my Model III, the Mark III and Mark IV packages, PRO-WAM, MISTER-ED, LS-DISKdisk, Little Brother, Little Brother Maintenance Utility, PRO-CREATE, DSM4, FASTBACK, to name just a few.

Besides the fine software products, Roy, your Programmers Guide, and distribution of THE SOURCE offer the TRS-80 far more insight than any of the CURRENT generation. Those who haven't availed themselves with the above publications would be well advised to obtain a copy while still available. Of course, if you NEVER intend to be more than a USER, purchase the manuals so that when you SELL your machine, the new user could use them.

I wish to clarify "UNSUPPORTED" software! Roy, you haven't been capable to release "UNSUPPORTED" software in the ones that I have run! I, in the future, would desire to replace the word "UNSUPPORTED" with "MATURE". After all, MATURE software typically has been running so long no one can remember the last bug that was corrected! The aforementioned software is MATURE. It remains true that some of the programs DON'T LIKE" one another very much, and that I must remember that I can't run ALLWRITE with ALTRES, I just add another BANK to my XLR8er ramdisk. Considering the "COMPLEXITY" the fact that crashes are not more often and more serious remains a credit to those who offer programs for sale.

Concerning the ongoing "TIME-DATE STAMP", the UNIX system that I worked on for a while, dropped the specific

"TIME-DATE STAMP" for DATE-ONLY. After 6 months, the year could be in field.

My background, I have been involved with computers since 1961, and have at times designed computer systems. I remain amazed with the power of even the 8-bit machines. Consider that the IBM System 360/30 was an 8-bit, the IBM System 360/40 was only 16-bits, and IBM System 360/50 only a 32-bit machine!

I currently work for General Telephone of the Northwest, and support as one function of my job, a digital switching system for up to 5000 lines based on the 8080 Microprocessor. How much power can we "UNDERUTILIZE"?

In closing, please accept MY BEST WISHES for prosperity and CONTINUED SUPPORT OF THE TRS-80 COMPUTERS.

Fm Hans de Wolf: I have some questions about some of the MISOSYS products, but first I must correct an error of mine. I wrote to you that 'Lair of the Dragon' (MEGADV1/MD3) could be made to run on the Model I by means of a one byte patch. This is wrong; the patch adjusts the address of the input buffer, so it must be a two byte patch (INBUF$ is 4225H in the Model III and 4318H in the Model I). The correct patch for MEGADV1/MD3 is

```
D50,91=18 43:F00,91=25 42
```

Now my questions: I am looking for a backup/restore program for my Aerocomp 20MB hard disk. Is it possible to use your RSHARD package for this, or are the backup/restore parts written especially for the Tandy harddisk? (the /DCT files probably will not work for the Aerocomp).

About the XLR8er board: I intend to buy an XLR8er from you and want to run LDOS 5.3, LS-DOS 6.3, and Montezuma CP/M. Do you sell driver software for all these environments or must I buy that from H.I.Tech? Do programs like SuperUtility Plus run on a XLR8'ed model 4P? (I know it is not your software and can't be responsible for it, but maybe you have had reports from other users).

A question about support: Do you support MISOSYS software that was manufactured and distributed by Molimerx (United Kingdom)? In the past I have bought from them your LC+EDAS package, which includes the SAID editor. This editor has become my favorite for general-purpose editing, but does not run under LDOS 5.3 because of the renamed keyboard driver module. The patch in the Quarterly was for a newer version of SAID, so I would like to upgrade it. By the way, is the name "SAID" an acronym for something? It has been always a mystery to me!

Final question: would you reconsider about binders for the manuals (the large size, 3-ring binders). I can understand that you sell not enough binders to make them commercially attractive to you, but in Europe it is a problem to find the right binders to put your software manuals in (European standard is 4-ring, and A4 size paper - just different enough to be a (minor) problem!)

Fm MISOSYS, Inc: Hans, Sorry, but I've been really busy here and correspondence has backed up. The archive/restore utilities that are part of the RSHARD package will work with any drive package - they're not driver dependent. The driver and formatter would not work, of course.

The XLR8er board comes with interfacing software for one DOS. We currently have software for DOS 6, LDOS, and MM CP/M. Each additional software set is $9.95. Once the board is installed, it should be transparent to all software - unless you have little low-memory available. The software interface in DOS 6 mode needs low memory. After installation, I can't see what problem there would be with any software unless it uses undocumented Z80 opcodes which are not supported by the 64180. TRSDOS 1.3 was one DOS which used some 8-bit loads into the IY index register. That was not supported by Zilog but nevertheless Z80 chips could execute such an instruction. The 64180 does not. TMQ had a patch a few issues back about that.

Now speaking of TMQ, Issue I.ii (Fall 1986) advised of the update to SAID and its cost. You could go that route, but at this point the cost is $10 (for a disk refresh) plus S&H ($5 in your case). Then the patches in TMQ for 5.3's KI/DVR would work. It also shouldn't take you too long to find the appropriate patch locations in version 1.0 of SAID. They really shouldn't be too far away from the locations in 1.1 release. And finally on the issue of SAID, the name was derived as a variation on the acronym of SED for Screen EDitor. But "SED" is pronounced as "said"; thus, SAID.

On the issue of binders, we got out of the large binder business for a number of reasons. Binders do not travel well. They're expensive to store. They tie up capital as they have to be purchased in sufficient volume to justify a reasonable price for imprinted custom binders. Freight costs are then magnified. Shipping containers are larger. You get the idea. Now we have gotten back into the binder business in the MS-DOS world. But there we use a small sized 9x7 binder which conveniently holds 8.5"x5.5" stock. We are also using that binder for LB and PRO-WAM. If I have the justification to reprint a Model 4 product, then I would consider shifting that to the smaller binder. So order PRO-WAM or LB and you can get a MISOSYS binder. I do still have some of the MISOSYS .75" binders (3/4") imprinted with our old address. If you want some of them, we used to sell them for $5.95 + S&H. A single binder weighs a little over a pound but packaged for shipping would probably be about 2 pounds. I have probably about 40 of them.

More printer woes

Fm Carol L. Welcomb, 11161 Edgerton NE, Rockford, MI 49341

After reading Mark Reeds' letter (TMQ II.iii), with all his problems with Okidata, I'm beginning to think I will have to fix this problem by myself! I'm hoping you (or any of your readers) will be able to help.

I have a Tandy 4P and DWP 220 daisy wheel printer. A friend gave us an Okidata Okimate 10 with a "Plug 'n Print" module for Commodore 64. This is a round, six pin male socket. Radio Shack (Linda Ebrahiem - Customer Relations in Ft. Worth) said if I had a Centronics parallel port, the Okimate 10 would work. The problem is the "Plug 'n Print" module. I wrote Okidata two months ago requesting a price for the proper "Plug 'n Print" module to work with my 4P. I do have a Centronics port!

The Okimate 10 is a quiet, color printer which will do graphics and I would really like to be able to use it. I could bypass the "Plug 'n Print" module altogether and create my own printer cable, but Okidata ought to support its systems. Do you have any ideas? Help!!!

On the price of DRAM...

Fm Ken Strickler: Roy, I am sure that you have heard the difference between a "Fairy Tale" and a "True Story", but to refresh your memory, a "Fairy Tale" begins Once upon a time... while a "True Story" begins, "Now this is no S*it..." - well:

Now this is no s*it. While I was a consultant back in 1975 for a manufacturer of an 8080 based computer system, along with several dozen other small manufacturers of computers, just as the computer field really began to ROLL, all of a sudden, it seemed that US small manufacturers seemed to be having trouble getting MEMORY chips at a reasonable price, if at ALL. (Sound Familiar?) I did some investigation at that time to see what the problem could possibly be.

Much to my surprise at that time I found that a 'LARGE', computer manufacturer, who shall remain "NAMELESS", had approached each of the MEMORY chip manufacturers and signed agreements to purchase their TOTAL memory chip output for the following 30 months! (Needless to say, COMPUTERS without MEMORY aren't your real HOT SELLERS!) The small companies folded as "DOMINOS", selling what assets that they had at FIRE SALE prices.

Now here we are again, MEMORY prices are climbing for no apparent reason, except possibly the fact that CLONE manufacturers can be getting TOO much of the MARKET, as the BIGS see it.

Now, of course, MONOPOLIES are prohibited, but who would stop a manufacturer from buying the CHIPS he needed? After all, it is just GOOD BUSINESS isn't it?

It seems to me, that the computer market is in for another shake up!

I'm sorry, but I had to get this off my mind, and just needed to write to someone who could possibly appreciate the gravity of the situation!

I have just "FIRED" up my RSHARD package, (my hard disk crashed about 2 months ago and I have just gotten it UP again) and would like to tell you what a fine package it is. Thanks so much for your continued support of this market.

I also see that 80 MICRO is gone, just got the last issue, and they deserved it!

Here is hoping that You, Brenda and the kids are all in GOOD HEALTH.

On computer mathematics...

Fm Donald W. Jones: Roy, I just got the spring issue of TMQ and I want to respond to your question "Should a number of articles on computer mathematics be written?" Yes. Yes. Yes.

Please count that as 3 votes. I've always been curious about how float and double floats are handled. I thought I saw an article once in 80-Micro, but I couldn't find it. Are floats handled in TRS-80 Model 4 Basic different than they are in MS-DOS basic?

Because of your informative teaching style, I would warmly welcome such a series.

On PRO-WAM upgrades...

Fm R. M. Doerr: What would be the cost to trade in my PRO-NTO on PRO-WAM?

Fm Norman P. Prosser: Please send Your latest catalog in support of the Model 3 and Model 4/4P TRS-80 computers. I received your 'REQUIRED READING' article and wanted to write to subscribe to your 'The MISOSYS Quarterly'. You will find a check for $25.00 enclosed. Please send my subscription to the address below. Also advise me on how I may get a copy of your back issues!

I wanted to tell you that 'Yes' I would be interested in a hard drive system for the Model 3 and Model 4/4P computers. The one described in your article for $495 sounds like a good price.

I purchased a copy of PRONTO and shortly after I received my copy I discovered that you had added enhancements and now call it PROWAM. This left me with a bad taste in my mouth. I know that enhancements are good and even expected but I would have preferred to have waited another month and received PROWAM than to have paid the same price for PRONTO.

The last thing I wanted to ask about was LS-DOS 6.3. I understand that there is a copy protection in this latest release and also understand that each user is supposed to purchase a new copy for each Model 4 or 4P that he or she may own. I have owned a Model 4 for years and have just bought a 4P. This is not feasible when you swap diskettes between the two computers. Also if I purchase the hard disk you described in your letter this would also cause problems because the LS-DOS 6.3 will recognize the computer as different from the Model 4 that I originally installed LS-DOS 6.3 on. I was told that when LS-DOS 6.3 is installed a snap shot of your computer is taken and if you try to run on a different configuration or boot a different computer the operating system will hang up. Please advise me on this issue because this will help me make up my mind about dropping 6.3 and going back to TRSDOS 6.2. My copy is LS-DOS63 LEVEL J SERIAL # B740I06264.

Fm MISOSYS, Inc: Norman, PRO-WAM Release 2 was announced over a year ago in our Spring 1987 issue of *THE MISOSYS QUARTERLY*. The name was changed from "**PRO-NTO**" to "**PRO-WAM**" back in mid-1986! When did you purchase your copy? And who did you purchase it from? When announced, we offered a trade-in program good through the end of March 1988. If you had purchased PRO-WAM 1.0 since April 1, 1987 you could have traded up to Release 2 for $20 + $5 S&H. Purchased prior to April 1, 1987, the cost was $24.95 + $5 S&H. We are still accepting Release 1 disks for trading up to the Release 2 for $39.95 + $5 S&H. If you care to take advantage of this tradein, just send your master disk back with the appropriate fee.

As far as your questions concerning LS-DOS 6.3, do not be alarmed over rumors. Changing your configuration will not affect the operation of the DOS.

Finally, a TMQ Index

Fm Elmar von Muralt: Roy, In my letter of the 30th January, 1988 I suggested to you to provide an index to TMQ. While I was browsing through all my TMQs looking for any bits and pieces on LB and PRO-WAM, it just struck me again how handy such an index would be. Well it was the Easter weekend with plenty of time at hand, so I decided to get stuck into it myself. Now I'm not claiming it to be complete or perfect, but it should come in handy all the same. I just hope you have not spent some of your valuable time on such an index already. It would be a shame having duplicated the effort. Possibilities I see are:

1. Include it in an otherwise thin TMQ (unlikely - there are no thin TMQs).

2. Post it separately together with a catalog & pricelist to all likely customers on your list (expensive)

3. Include it with any of your software packages sold.

4. Put in onto DISK NOTES in the LB file (limited circulation).

As indicated in brackets all options have some drawbacks but I feel it is quite handy to have this index when looking for those "I know I've seen it somewhere in TMQ" situations, so why not share it around somehow. Of course I leave it up to you if and how you are to publish this index, as it is rather lengthy. Also I left out any patches from the Patch Corner as I am not informed enough to establish which patches may supersede previous ones. So you still may need to organize an index of all current patches.

While I was working on this index I have overcome the skipping of forms problem I mentioned in my last letter. Originally I had set: FORMS(PAGE=70,LINES=60) to accommodate A4 forms and 5 line top and bottom margin. LB's printer parameters were also set to these numbers. I suspect both LB and in turn FORMS generated their own TOF code, i.e., 2 TOF codes were sent to the printer, thus skipping a form with each page. The cure proved to be to set FORMS(PAGE=70,LINES=70), so FORMS won't generate its own TOF. (Just scanned the index to find the reference which explained it, not under LB but under Model 4 II.iii.p.58, so what - can't win'em all!)

Another curious thing happened while FORMS and LB had the same number of lines: From page 2 onwards on each line No. 28 the first character of the first field was not printed and the rest of the line appeared shifted left by one character position except for the Rec.# in the most L.H.column which was properly aligned. I checked out some other printouts I had from my stocklistings and the same thing happened but on some listings the error occurred on line 29, on other listings on line 30 but consistent within the record.

Sometimes the fingers are too quick to hit the <F3> (Save) key while adding new records to LB files. I just wonder if Options 2 (Add) and 3 (Update/Delete) could be combined. Alternatively is there an automatic way to initialize new records as "empty" but accessible records? I can do it manually by just saving a lot of records without any data being entered in the Add mode and then Add or Edit in the Upgrade Mode. Unfortunately this method produces

unsatisfactory printouts as empty records are being listed unless all the prepared "empties" are deleted prior to printing.

In response to your invitation in TMQ II.i.p.8: "Just what do you all do with your systems?", here is my little bit. Back in '82 we bought an NBI System 3000 Wordproccessor with a Ricoh RP1600 Daisy Wheel printer. It was to help us with our invoicing with the "Stored Keystrokes" and "Stat/Maths" package I got a System going which allowed most of our daily tasks. Unfortunately two of my clerks who each had been sent to Remington, the Company selling NBI in Australia, to be trained for a week decided they get better pay in the Public Service and quit on short notice. Well that was the Governments gain. The next clerk was reluctant to use the WP but came up with a very simple "manual" system which required only single sheet Invoice/Jobcard and we purchased a photocopier for the few cases when copies were required. However, by the time the WP was "retired" new models had been released and we couldn't sell it anymore for anywhere near its worth. So we took it home and my wife Madeleine registered her own business to do contract wordprocessing. She done a fair few reports for consultants and generally enjoys the type of work she gets. The only drawback with the NBI is the lack of a spelling checker. Meanwhile Madeleine has started a full time job at the same school where Alex and Nathanael go to where they got MACs. Having worked on their WP package she doesn't mind at all our good old NBI.

In 1985 I was approached by an elder of our Lutheran congregation to become treasurer. After checking what was involved with the previous treasurer I accepted and took that as a good enough reason to get my own computer. A friend of mine is the local Tandy dealer and he recommended the 4P, I believe it was the last one he sold! Never mind, he threw in Visicalc too, so I was quite happy and soon had the ledger and various other reports organized to keep track of our congregations' financial affairs. Then I heard about Multiplan, so I bought it and set up my business ledger with it. At first I thought it was my non familiarity with the product, but after two years of battling with its slowness and single-key command structure I've come to loathe it. If it weren't for the sort capability I would have discarded it long ago. I still do the church work with Visicalc and, antique as it is, it is a breeze compared to the supposedly superior Multiplan. Then I wrote my own BASIC programs to record, edit and list stock records, however, it was a rather laborious task as nearly every report required a separate program. LB has come in very handy there and all my stock records are now converted to LB files - Gee, that sort works like a breeze - Thank you! That leaves this contraption of a "wordproccessing package" SuperScripsit 1.01.00. It does silly things at times with just the <Shift>-key depressed: it may delete parts or all of the current document or add on the contents of some keystoring buffer - all rather disconcerting, especially after having been spoiled by the workings of the NBI.

This leads me to the next question: Which wordprocessing package can you recommend that works on LSDOS 6.3L and doesn't drive me around the bend?

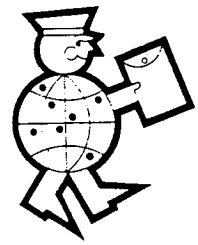
What are the requirements of a Modem which should work now with the 4P and later on with an MS-DOS type machine?

Just a couple of comments about the presentation of the last TMQ (II.iii): Cut down on the number of different fonts. Page 15 has at least 7 different fonts just in the text and not counting italics, add the headers and footers and you have 10 fonts. Fantastic capabilities of your printer, but don't overdo it, keep it to say no more than 5 fonts. Of equally low level disturbance were some inconsistencies of upper and lower case titles. Having said that I'm looking forward to the next TMQ but even more to the XLR8 board which I'm ordering today.

What's all the fuss about having to get a new DOS (II.iii.p.11)? Thanks to the expiry of TRSDOS 6.2, I had to get LS-DOS 6.3, as advertised in 80 MICRO, definitely the best thing I got out of it - I'm still waiting for back issues I had paid for!!! Thanks to LSI I was pointed towards MISOSYS. I don't know if that was as a matter of course or because I asked for info about any other software they might have. Well now I subscribe to TMQ and get patch reports, problem discussions and much more on a regular basis and it's presented in a form that makes me feel more like being a member of a club rather than just a subscriber to yet another publication. As for the new DOS, i.e. LS-DOS 6.3L, alone the changes of the editing features in BASIC are worth the whole price. How much more valuable is the access to SVCs for those programmers who want to get a bit closer to the machine. Then there is TED the ASCII Text Editor - worth the price of the whole disk - and I haven't even mentioned BREF or DISKCOPY.

What were your reasons to choose C over PASCAL, or any other language for that matter, as your software development language? When comes the point when BASIC just won't do anymore? What texts do you recommend teaching how to program in C? When should I rather choose RATFOR as my next language? A pity you don't sell a FORTRAN compiler to go with it.

Enough of all that, else you might accuse me of having an acute bout of verbal diarrhoea. All the best to you and your family, I hope all is well with Brenda and your baby.



DOS Subjects

LDOS 5.3

LDOS 5.3's TED editor

Fm A. J. Paszyc: I have been using your LDOS 5.3.0 for one full year now, and generally I am well satisfied with its performance. I noticed, however, that from about 9 p.m. the time stamp becomes incorrect.

More recently I also found out that the TED editing commands do not work. Specifically, I can invoke TED after installing the KI driver by using TED [filename/txt] command, but cannot load a file using <CLEAR L> command after TED has been previously installed.

When the file is in the buffer, I cannot toggle "overstrike" to "insert" mode using <CLEAR A> command, neither can I specify "DELETE" with <CLEAR D> command.

I am using a Tandy 4D machine which has the <CLEAR> key, but I also tried to use the <SHIFT><DOWN ARROW> as control key but it does not work either.

The README/TXT file included on my disk shows a number of corrections made to LDOS 5.3.0 including one or two to the TED editor. I wonder if these corrections were inadvertently not made on my disk. Or maybe I am doing something wrong. Please help!

Fm MISOSYS, Inc: A.J., I believe the answer to your question is right there on page 15 of the LDOS 5.3 documentation update. "Most of Ted's commands are entered with the control key (<SHIFT><DOWN ARROW> under LDOS). ... the control key will be noted by the caret character, '^'."

Your letter stated that in every case, you were trying to use the <CLEAR> key, not the CONTROL key! Even though a Model 4 has a CTRL key, unless you are using the KI4/DVR which is part of the Model 4 Hardware Interface Kit (M-12-110), the LDOS KI/DVR recognizes only the <LEFT-SHIFT DOWN-ARROW> combination as CTRL.

Note the following text which is part of the README/TXT file which is on the LDOS master disk. It was added back in February of 1987.

Note: If you are using a Model 4 with clustered ARROWs which are adjacent to the <RIGHT-SHIFT> key, you can apply the following patch to have KI/DVR use <RIGHT-SHIFT> <DOWN-ARROW> for use as the control key instead of <LEFT-SHIFT> <DOWN-ARROW>:

```
PATCH KI/DVR.DRIVER
(D01,EC=4E:F01,EC=46)
```

I believe this should put you in the correct direction to make full use of TED.

SuperUtility and LDOS 5.3

Fm Robert E. Murch: Roy, Does it make sense that there is a minor incompatibility between LDOS 5.3.0 (TC0A0524) and SuperUtility 3.2 (#84308)? I think what happened was that I repaired the HIT and GAT sectors of a LDOS 5.3.0 data disk and did some file copying using SU and then had a series of problems back in LDOS 5.3. I didn't Capture all of the detail but the data disk in drive 1 wasn't recognized from time to time and I finally discovered that rebooting made drive 1 available.

Trying to backup with the "X" option was not successful -- a "directory read error" was the problem most of the time. And finally I noticed that the "time" field in the directory listing was blank. I did a "dateconv" on the data disk and I think things are back to normal, although I don't think the "X" backup works yet.

Any thoughts? Necessary to do "dateconv" after using SU do you think?

Fm MISOSYS, Inc: Robert, I do not know what SU does in its "repair" of the GAT and HIT sectors. Although I have documented the changes to the directory structure of LDOS 5.3, I doubt that SU was revised. Thus, I cannot speculate on what needs to be done after you use SU. Obviously, if the TIME field was blank, LDOS no longer recognized that disk as a 5.3 disk. SU probably rebuilt the configuration byte and didn't set the "extended date" flag. Your best bet would be to copy the files off on to another disk then re-format the "bad" one. You can't just DATECONV because that will clear out

all of the extended date/time fields of each directory record with a subsequent return of 1988 dated files to 1980.

Alpha clock patch

Fm Stu Martin: Roy, Here is a copy of the ALPHA clock patch for LDOS 5.3. The only problem noted, during the last year, was when the clock is removed from the system. When removed and the system booted, the date and time is invalid. (i.e. garbage). I've noted this within the patch.

I haven't had time to look into this since it works fine with the clock installed. I have not had any problems running any Model-III program to date but note that I'm running mostly Model-IV. I hope readers of TMQ will continue with this patch and pass along any feedback.

Thank You and best regards, (for a great publication)

P.S. I've included a copy of the original 5.1 patch for Reference.

```
.Newclock/80 fix for LDOS 5.3.x only.
.by Graham M. Brown, updated for 5.3 by
. Stu Martin .(Adapted from Roy Soltoff's
. Model-I zap)
.
.Note: This is an temporary patch. It
. works fine but if the ALPHA clock is
. removed from the system, this patch
. should be removed as well. If not, the
. date will be invalid after .booting but
. can be corrected via the date/time entry
. at the LDOS Ready prompt.
.
.PATCH SYS0/SYS.SYSTEM LDOS53 (OPTION=OFF)
.
D0D,94=3E 10 D3 EC 21 1C 42 01 BA 01 CD D2
D0D,A0=4E 06 03 CD D2 4E 01 BC
D0D,A8=0F CD D2 4E DB BB E6 07
D0D,B0=47 E6 03 11 1B 42 21 18
D0D,B8=01 02 CB 4E 44 20 17 CB
D0D,C0=FE 18 13 ED 78 0D A0 07
D0D,C8=57 07 07 82 57 ED 78 0D
D0D,D0=E6 0F 82 77 2B C9 00 00
D0D,D8=00 00 00 00 00 00 00
D0D,F5=1A 42
D0E,25=00 00 00
. eop
```

```
.Newclock80 fix for LDOS 5.1.x for the
. Model III by Graham M Brown
.
.Allows the DATE to be read from BOOT-up
.
.(Adapted from Roy Soltoff's Model I zap)
.Apply this to SYS0/SYS
.
D0D,9A=3E 10 D3 EC 21 1C 42 01
D0D,A2=BA 01 CD CA 4E 06 03 CD
D0D,AA=CA 4E 01 BC 0F CD CA 4E
D0D,B2=DB BB E6 07 47 E6 03 11
D0D,BA=1B 42 21 18 44 20 23 CB FE 18 1F ED
D0D,CA=78 0D A0 07 57 07 07 82
D0D,D2=57 ED 78 0D E6 0F 82 77
D0D,DA=2B C9
.EOP
```

LS-DOS 6.3

What's 6.3's latest Level?

Fm Patrick H. Larkin: Roy, Enclosed please find completed registration card for my second copy of the LDOS 5.3 upgrade kit.

I was pleased to note in your "Required Reading" cooperative Mailing effort that you are now supporting LS-DOS 6.3. Will you be printing the patch(es) necessary to upgrade from Level K to Level L? (I always wondered why LSI never provided it to you, or if they did, why you never printed it.)

Now, I have a few questions -- do you know whether Model 4 VisiCalc will utilize additional RAM provided by the XLR8er board? I called H. I. Technology several months ago, and the gent I talked to didn't know, but mentioned you would be discussing it in some detail in TMQ (which you did), and suggested I either ask you or send them a copy of the program. I'm trying to decide whether to upgrade my old (B&W video, separated arrow keys -- upgraded from cassette only, I believe) Model 4 (I bought it used). I have found that many of my VC spreadsheets expand to utilize most of the memory, and if I have SPOOLER active I sometimes don't have enough available memory. If additional RAM available with the XLR8er will be utilized by Visicalc AND calculations will be sped up, I would seriously consider upgrading with the board. Also, according to something I saw in 80 Micro, if one happens to get too large a VC file with SPOOLER active, it will (can) cause a system crash. Does the SYS8 patch on page 96, TMQ II.ii, have some bearing on this, or does it only apply to PRO-WAM with ALLWRITE and the SPOOLER?

Another Question -- do you know of a fairly decent bookkeeping system that will run on the Model 4 (either mode)? I tried The Home Accountant, and found it to be VERY slow (at the time I was running my Model III with TRSDOS 1.3 - I hadn't seen the light). I don't know whether that was because it was written in BASIC or because it was running under TRSDOS, but boy, did I get tired of waiting! In that vein, do you know anything about the CASH PROFESSIONAL by Microdex? For that matter, what accounting system did you use before you switched to MS-DOS?

Keep up the good work on LDOS and TMQ - I'm sold on them!

Fm MISOSYS, Inc: Patrick, LSI has provided me the input for Level K to Level L changes. But the input was in the LSI Column printed in TMQ II.iv which was mailed on May 27th. As an aside, the very latest release of LS-DOS 6.3 is termed "Level L+". It has been that way since February of 1988. LSI released 6.3 on January 2nd, 1987. It was then at Level H. On February 2, 1987, LSI applied patches 1-9 and re-assembled BREF. That release was designated "Level J". I really don't know what happened to "I"; maybe they omitted that to avoid confusing it with a "1". The applications of patches 10-16 brought it to "Level K"; patch 16 was dated 7/1/87. Apparently, when LSI redid their master at this point, they posted it as "Level L". They did this to distinguish a disk patched by you and one prepared by them. I think it just caused more confusion. Patch 17 was applied on 11/1/87; this changed LSI's designation to "L+". MISOSYS published patch 17 in TMQ II.iv as noted above.

I doubt that Model 4 Visicalc would utilize any extra RAM even if supported through @BANK as is available via the XLR8er board software interface. Only a very few companies were astute enough to recognize the possibility of additional RAM. Tandy was not!

On the other hand, you could run the spooler using a bank higher than 2. The spooler as written will actually accept a bank value of 1-7. Even that could be easily patched to accept 1-10 for total support of any bank available with the XLR8er. And yes, you could crash in strange ways if that patch to SYS8 as discussed in TMQ II.ii is not applied. It doesn't just restrict itself to ALLWRITE and PRO-WAM.

I can't recommend a bookkeeping system for the Model 4 as I never used one on the 4. Prior to shifting over to Bedford Accounting on the PC, I used a very old Model III based package which was based on Osborne & Associates General Ledger package. A lot of string-driven BASIC programs wind up being slow due to excessive time spent in string space manipulation (garbage collection). I also am unfamiliar with Microdex's program. Perhaps I can throw the question open to TMQ readers.

Help with Model 4 products

Fm Walter Royal Jones, Jr.: Friends, I can't tell you how pleased I am to get your latest mailing. Enclosed in a check ($30) for MISOSYS Quarterly.

Since I started out with Model III, later adding a Model 4 (128K), I have a few programs (Homeacct, Profile III+) that currently run very well in Model III mode, but which I would dearly like to run in LS-DOS instead, for its faster speed and other features. Does anyone have anything which can facilitate such a program upgrade?

Failing that, Is there anything for Model 4 comparable to Homeacct? And would records be transferable?

How does LB compare with Profile IV? (assuming that is still available) Can you create your own menus? Can you scan records in both directions? Can you transfer data to Scripsit-Pro? Is there an internal math package? Would it function with Tandy's Double Duty program? (or would the demands on RAM exceed 128K?)

The XLR8er sounds alluring, but as a soldering iron sub-bungler who might be capable of opening the case but that's about all, I have misgivings. Exactly what is meant by "simple plug-in installation"? Plug in to what? Does the Model 4 have room for "cards"? (And what effect would such accessories have on the Tandy service contract?)

PRO-WAM in its earlier incarnation used to include a function which would allow use of the printer like a typewriter. I note that this is not mentioned in the current advertisement. What happened?

I hope all this doesn't sound too picky. What is really going on in my mind is an unresolved question - whether to stick with the Model 4, which I like, and with which I am familiar, or whether to abandon all hope (not to mention software and documents), and go MSDOS for the sake of compatibility, more power, etc. Having looked at several MSDOS options, I have come to realize how good the Model 4 is, and how comfortable I am with it. But if it and its documents are soon to become dinosaur fossils?

Fm MISOSYS, Inc: Walter, Let me try to answer your questions, some of which actually require great depth which I really cannot go into here.

It would take a huge amount of patching, if even feasible, to convert PROFILE III+ to Model 4 mode. I know of no one who has tried to attempt that. Actually, if you had the PROFILE III+HD version, it would run under LDOS and then could be used on a 4 in III mode at the faster speed. LDOS at high speed is actually faster than DOS 6 at high speed because LDOS does NOT have

to bother with bank switching memory management. But the HD version of PROFILE may not be available from Tandy.

I am not familiar with HOME Accountant; thus, I cannot advise you as to what is available on the Model 4 side which is compatible.

I am enclosing a small brochure which gives some more information on LB. Specifically, LB probably has most of the features of PROFILE III+; but may, in fact, be easier to use. You cannot create "menus" but you can create automatic job files which can then be invoked. We use this facility to automate re-indexing a database. Finding a record is performed sequentially when you are not using an index. An indexed search is performed via a binary search algorithm. At any time you can step forward or reverse through the data base either with an indexed sequence or straight sequential. If you call the "single stepping" scanning, then yes, LB supports that. Data cannot be transferred to Scripsit as part of any procedure currently supplied with LB. But LB does have limited form-letter capability. LB records support calculated fields with 4-function capabilities (add, sub, mul, and div). It could function with DoubleDuty if you are running with either a hard drive or double-sided drives. On a standard 2-drive 128K Model 4, LB requires that the system drive be the Memdisk.

The XLR8er requires no soldering. The plug-in mode arises from the fact that it plugs into the Z80 socket. The XLR8er replaces the Z80 CPU chip with a faster enhanced HD64180 processor. The XLR8er board mounts onto a new shield which is supplied by us. The shield replaces the existing shield around the motherboard of the 4. And no, the 4 doesn't have any facility for plug-in cards (the 4P does have one slot for a modem card). If you have a Tandy service contract, I would guess that you would have to revert the machine to "stock" before seeking repair. Tandy doesn't like to bother with add-ons.

The current PRO-WAM still includes TYPER. We just don't have the room in an ad to mention everything which comes with the Release 2 - it is an extensive enhancement.

My philosophy on staying with the Model 4 or moving to MS-DOS (or other) is simple. If your 4 is serving your needs, don't jump to some other architecture. Move to higher power if you must acquire it to run some particular new application now. For instance, if you have a need for desktop publishing (DTP), you may have to resort to either a souped-up Macintosh or an AT-class PC (or higher, i.e. 386). Of course, you need to evaluate additional investments to your current setup. The best bet to consider, is to add peripherals which are re-usable in other environments. That's why we are designing our Hard Drive package to be totally re-usable under MS-DOS.

Typical 6.3 Support Questions

Fm MISOSYS, Inc: Since MISOSYS has taken over the complete and total sale and support of the LS-DOS 6.3 product on June 1st, I have begun to reckon with the users Bill spoke of some issues ago. Knowing how some folks took offense at the words he used to describe these computer "users", I will refrain from making generalities. On the other hand, I would like to provide you readers with a sample of the kinds of support requests I have received of late; some reasonable, some not so reasonable.

1. How do I gain access to drives 2 and 3?

2. I just moved up to 6.3 now SCRIPSIT won't center on printing.

3. How do I get rid of system files? REMOVE doesn't do it.

4. I have 6.3 but my SuperScripsit disk doesn't accept the new date.

5. How do I format a 15 Meg hard drive?

6. Where is MAILLIST/BAS?

7. FORMS filter is converting CR CR to CR LF. [Note my statements which follow concerning this one.]

8. Are 11H errors common?

9. I'm getting a read error in copying a file.

10. Characters wave on the screen.

11. How can I update my Multiplan disk?

12. I can't BACKUP my deskmate disk.

13. I have a problem with Scripsit.

14. VIDTEXT gives "Communications driver not loaded".

15. I'm getting "Program not found" when typing RM.

16. How do I remove protected files?

And on and on. I think that 90% of the calls stem from folks totally unwilling to read their user manuals. Bear in mind too, that a lot of the above statements came after

trying to extract an intelligent error report out of the caller. The request I would make of my readers is to urge all other users to to WRITE down the sequence of steps taken and the responses displayed, if any, when a problem is being experienced. It serves no purpose to call the next day with a report, "I can't back up my disk", or "my deskmate program won't work", etc. When I ask, "Why not, what error message did you get" and you report back, "I don't remember", I think you get the idea. I'm sure that most readers of TMQ do not fall into this category. But you may know of others who do. Advise them to produce detailed trouble reports. I have a hard time helping anyone who cannot explain their problem.

Turning now to the FORMS filter report, I had a report from a 6.3 user who was using the FORMS filter for some massaging of data. By ROUTING a device to a file, then filtering that device, copying a file to the filtered device would produce a "filtered" file. Unfortunately, the filter was doing one thing unexpected. If two 13D characters appeared in a row, the filter would change them to the sequence, 13 10. In other words, a CR CR sequence was converted to a CR LF sequence.

So I dug into the filter code to investigate that report. On page 211 of THE SOURCE, there is some code starting at the label "DOWN1". This code handles the case of a return output as the first character on a line; i.e. a text stream which consists of a blank line. Old time TRS-80 users may recollect that back in the old days, there were some printers which would not do a line feed if a RETURN was sent and the print head was at the first column. There are probably still some of those printers in use. Why did thousands of BASIC programmers code PRINT " " when a simple PRINT was supposed to do a line feed. Why because some printers wouldn't advance to a new line under that circumstance. That's why the FORMS filter is coded to convert a CR to a LF if the character counter is zero. If you want to deactivate that feature because you don't want CR CR sequences changed to CR LF, just apply this little patch:

```
PATCH FORMS/FLT.FILTER
(D03,31=18:F03,31=20)
```

MS-DOS

PageView; Forget it!

Fm MISOSYS, Inc: Here's an open letter to Microsoft which addresses our frustrations over the limitations of their "new" PageView product.

Dear Microsoft Product Support,

This letter is in response to hours of frustration over your Pageview product which I purchased under the assumption that it was going to allow me to integrate graphics into my WORD processing. What a false assumption. Let me explain the problems I have experienced.

I'm using an AST Premium/286 computer with MS-DOS 3.3. The printer is a NEC LC-890 Postscript capable printer. I'm using Microsoft WORD version 4.0 and Microsoft WINDOWS version 2.03. Pageview is release 1.0.

First, since Pageview and WORD cannot both open the same document simultaneously, the frustration which arises from having to clear the document when switching from one to another makes PageView practically unusable. But that's the least of the problems.

I spent many hours trying to understand why what I saw on the PageView screen was not formatted anywhere like what was formatted by WORD. I am set up using the POSTSCRP.PRD in WORD and am also specifying POSTSCRIPT printer in WINDOWS. Okay there. Most of my word processing is for a magazine which I publish. The primary font is Times Roman. I also make use of boxes and lines which require SYMBOL.

After much puzzling and printing, I determined that Pageview was using Courier where I specify Times Roman in WORD. So I called the Microsoft Support organization and was asked if I created the PFF file for my PRD file. I stated that the POSTSCRP.PFF file was supplied on the PageView disk. This was a Microsoft-provided file. In any event, the support individual suggested I examine that file for correctness!!!!! So I did. Following the book, I brought up WINDOWS-WRITE and wrote down the names of the supported fonts. I then brought up WORD and wrote down the names and generic identifications of fonts supported by WORD with the POSTSCRP.PRD driver. I then went to the PageView Appendix A table and correlated the PRD font numbers with the generic font names. It's funny that the Pageview manual states, "*Type each numeric font code, a semicolon (;), and the Write font name that corresponds to the numeric code.*" Yet, the POSTSCRP.PFF file supplied on the PAGEVIEW disk is as follows:

0;Courier, 1;Courier, 8;Helv, 9;AvantGarde, 10;HelveticaNarrow, 16;Bookman, 24;TmsRmn, 25;NewCenturySchlbk, 26;Palatino, 27;Souvenir, 50;ZapfChancery, 56;Symbol, 60;ZapfDingbats

Those font names are supposed to be the same as WINDOWS Write? I see "Courier", "Helv", "Symbol", and "Tms Rmn" showing up in the Write screen. Why are all those other names being used in the PFF file provided by Microsoft? Under close scrutiny, I noted that Write had "**Tms Rmn**" whereas PageView provided "**TmsRmn**". That single SPACE was quite important. After I edited the PFF file provided by Microsoft, and changed "TmsRmn" to "Tms Rmn", I came

very close to seeing the PageView screen with the same formatting as that generated by WORD.

I noticed two things which caused a difference in formatting between WORD and Pageview. One was an extremely small difference in the calculation of character widths. This caused the last word on a particular line in WORD to be bumped to the next line in PageView. The second thing I noticed was that PageView did not separate a hyphenated word that WORD did. This, of course, caused page formatting to differ from that point on.

Well, I accepted slight differences and proceeded to attempt an insertion of a picture frame - or I should say picture frames. I first discovered that I could not insert a picture frame immediately adjacent to another picture frame. That's one of the things I wanted to do. Thus, another drawback. The final blow came when I went to print the result of my labors. The pictures didn't show up nor did the box surrounding a heading.

I am enclosing a 360K diskette which contains the BLURB.DOC file, the associated .STY file, the picture files, the modified .PFF file, and the POSTSCRP.PRD file. I am also enclosing copies of the pages referenced in this letter. For now, I find Pageview totally unacceptable. It doesn't print the pictures, it doesn't print the boxes, it doesn't come close enough to the formatting generated by WORD. Thus, it is incapable of aiding me in incorporating pictures with my .DOC files. Unless the formatting of WORD and Pageview are identical, Pageview cannot be used.

My alternative is to dump the $50 program and hope that Microsoft introduces a usable page display adjunct to WORD, keep Pageview and expect to receive a free update which fixes these problems, or shift to WordPerfect version 5 which incorporates the page display viewing in the word processor itself - where it belongs, not to mention adds better control over mixing multi-column formats on a single page.

How about it folks, do you have some concrete solutions to these problems?

Here's help for the Tandy 1200

Fm David Goben: Roy, Again I write. So soon? Actually, I did not wish to mix too much stuff up with my order, so I chose to send this separately.

In regard to Bob Haynes on page 36 of TMQ II.iii, and Hardin's reply. If Bob does as Hardin Suggests, and goes to MS-DOS 3.2, then he will first have to upgrade his ROM from 2.5 to 3.0. Radio Shack supplies this service for around $35, which includes MS-DOS 2.11.41, BASIC, and Manuals (Bob may find that some 2.11.41 applications and utilities, such as FORMAT, will not work properly with the 2.5 ROM). Also, the 3.0 ROM allows the 1200 not only to continue to be fully XT-compatible, but also fully compatible with the TANDY 3000 and 4000. Thus Bob can purchase the TANDY 3000/4000 MS-DOS/BASIC systems, such as 3.2, and operate with them without problems.

However, using the 3000's DOS on the 1200 will cause the floppy drive tracking to sound differently, giving a courser sound while moving between tracks. He can smooth that out easy enough. I don't understand why this doesn't work from a COM file under 3.2 (it works fine under 3.1), but my method runs OK from DEBUG:

From the DOS prompt (A> or C> or whatever), enter the following lines (do NOT include the comments to the right):

```
copy con db
a                <-- go into the ASSEMBLE mode
int 13           <-- reset floppy. In this case AH is
                     already zero
int 20           <-- short and dirty exit which works
                     well from DEBUG
                 <-- this is a BLANK line. Just press
                     ENTER
g                <-- execute the small program
q                <-- leave DEBUG and return to DOS
```

Close up the file by pressing F6 and then ENTER. Now include the line DEBUG <DB in your AUTOEXEC.BAT file, preferably as the first entry, or simply enter DEBUG <DB from the DOS prompt. This process will smooth out disk track stepping considerably under MS-DOS 3.1 or 3.2.

On another subject, in Bill's column, page 31: column 2, paragraph 6, he said that MS-DOS would not accept the year 2000. This is incorrect, for it will accept dates of up to 2099. Instead of typing 00 (zero-zero) as he demonstrated, type in 2000. MS-DOS will accept full year entries, unlike the TRS-80 systems, which, will only accept the last 2 digits. The date field reserves 7 bits for the year, relative to 1980. Although this allows for 128 years, for some reason MS-DOS allows only 0-119, providing for 1980 to 2099.

The SmartWatch in my 1200 will only display correct dates up to December 31, 1999, however. After that, even though it will show a fluky date in the year field when you invoke the SmartWatch, it will still install it correctly in the system storage area, so that if you typed DATE afterward, DOS will show the correct year information.

Hope that this information can be of use.

Roy's Technical Corner

Getting into computer math by Roy Soltoff

For quite some time now, I have dealt with reports about inaccurate calculations in computer programs. The reports generally relate to the use of single or even double precision floating point numbers. Try as I could, in a few brief statements, or even paragraphs, I had a hard time expressing the fact that computer-processed floating point mathematics was inherently imprecise, and that a computer could do wonderful calculations for you once you knew and understood its limitations, and dealt with them. That, of course, is a most difficult topic. The error analysis of floating point math is really beyond the scope of most folks, and certainly beyond anything I could posture in this publication.

On the other hand, I have come to realize that perhaps its time to expose the TMQ readers to the intricacies of computer math. I dare not jump right in at the floating point level; because to do that would thoroughly confuse a good percentage of my readers. Let me then start out at ground zero. Let me also relate that I am no expert. I just happen to have spent some time in developing floating point routines. My background in Engineering has exposed me to a great deal of mathematics, and my post graduate work in Operations Research was nothing but mathematics. Mathematics is also a subject feared by many. But I hope to bring most of this discussion down to earth. Bear with me. Let me know when I go off base.

Now an approach to computer math must first come with the assurance that we all understand the kind of mathematics we deal with daily, and the kind of numbers we work with. Our system is termed a *positional* system, because the position of each digit in a number has something to do with the value of that digit. What other kinds of systems are there? Well at a minimum, there are non-positional systems. The Roman system, which by the way is used to designate the issues of *THE MISOSYS QUARTERLY*, is an example of such a system. In that system, symbols are used to designate particular values. The following table illustrates some of these values.

I	1
V	5
X	10
L	50
C	100
D	500
M	1000

In this system, a number is a string of Roman numerals. The number value is figured by adding up the values of each symbol from left to right; if a symbol has a smaller value than the symbol to its right, then it is subtracted from that symbol's value and the result accumulated into the addition. For example, the Roman number, **MCMLXXXVIII**, represents the current year expressed as "1988" in decimal.

But let me get back to our number system, especially since I just mentioned the term *decimal*. In any positional number system, any number can be represented by the general formula,

$$(...a_3a_2a_1a_0.a_{-1}a_{-2}...)_b = ... + a_3b^3 + a_2b^2 + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + a_{-2}b^{-2} + ...$$

In this notation, the "a's" represent positional coefficients with the subscripts designating the position; and "b" represents the *radix* or *base*. The period which you see between the "a_0" and "a_{-1}" is called the *radix point*. The term "base" may be more comfortable than the term "radix", but both are equally correct; in fact, "radix" may be the term mathematicians prefer. In any event, the "base" is the number which is raised to a power in order to arrive at the magnitude of the position. The superscripts indicate the power to which the base is raised. If the base is 10, the number system is termed *decimal*. Two other terms can be introduced here. The *most significant digit* is the leftmost or leading digit. Conversely, the *least significant digit* is the rightmost or trailing digit. The terms arise from the fact that digits which are coefficients in positions where the base is raised to a power greater than other positions have more significance; they are "more important" to the effect of the number, so to speak. For instance, If I say I just won $10,001 in the sweepstakes, that leftmost "1" represents $10,000 while the rightmost "1 represents $1; certainly the $10,000 has more importance to me than the $1.

hexadecimal is properly termed sexadecimal

Other systems commonly in use are: *binary*, base 2; *octal* (which some say should be termed octonary or octonal), base 8; *duodecimal*, base 12; *hexadecimal* (which experts say is properly termed sexadecimal), base 16; and *sexagesimal*, base 60, commonly used for calculations in geodetics (degrees, minutes, seconds,...).

If we confine ourselves to a three-place decimal number, say 473 for no particular reason, and apply the general formula, we have a result like this:

$$\begin{aligned} 473 &= 4 * 10^2 + 7 * 10^1 + 3 * 10^0 \\ &= 4 * 100 + 7 * 10 + 3 * 1 \\ &= 473 \end{aligned}$$

I have used a "*" symbol to denote multiplication. One of the restrictions on a number system, is that no digit can be of a value which exceeds the base. Thus, we have the class of symbols, {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} representing the ten digits making up the decimal system. Similarly, if we were to consider an octal system, its eight symbols would be the class {0, 1, 2, 3, 4, 5, 6, 7}. Likewise, the binary system would use a class of two symbols: {0, 1}. Hexadecimal, being base 16, has 16 symbols. We arbitrarily use symbols of the alphabet to indicate digits of value greater than 9; thus, {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F} are the class of symbols for hexadecimal.

Very early computers actually were decimal based. I do recollect some early work towards a ternary-based computer using tri-state devices; however, since digital electronics oriented towards a scheme of gates which had two distinct "positions", computers soon switched to the binary system. That's why its important to understand the binary number system and binary mathematics.

The binary system is simple since there are only two symbols in its class, {0, 1}. If we want to write the number "01011011" in our general formula construct, it would be:

$$0*2^7 + 1*2^6 + 0*2^5 + 1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0$$

When dealing with the binary number system, a power of two table is convenient to either have, or remember. It's quite easy to produce by calculating (if you know how to multiply by 2 without a computer). Here's a short power of two table:

2^0	1	2^8	256
2^1	2	2^9	512
2^2	4	2^{10}	1024
2^3	8	2^{11}	2048
2^4	16	2^{12}	4096
2^5	32	2^{13}	8192
2^6	64	2^{14}	16384
2^7	128	2^{15}	32768

This shows our number has a value of:

$$1*64 + 1*16 + 1*8 + 1*2 + 1*1 = 91D$$

Here I have introduced additional notation. The suffix, "D", designates the base of the number. Without that suffix, how would you know I wasn't talking about a hexadecimal number? You wouldn't. Therefore, whenever we are working with different number systems, it is important for us to designate the base of a number by means of a suffix symbol.

We commonly use "D" for decimal, "B" for binary, "H" for hexadecimal, and "O" for octal. We generally also assume that a number without a base suffix is expressed as decimal. Sometimes when writing in certain computer languages, its acceptable to use a "Q" as a designator for octal to avoid confusing the letter "oh" and the number "zero" when reading the numeric value. Other problems exist when the language assumes hexadecimal as the default and you want to write a hexadecimal value ending in "B". Is the number "10B" hexadecimal or binary? To avoid even that confusion, some languages use a "V" to indicate binary; but that really isn't too common.

Up to this point, you should have recognized that the way in which we convert a number from one base to decimal, is to add up the positional values according to our generalized formula expressing the number. But how about converting a decimal number to another base? We can do that easily by repeatedly dividing the decimal number by the desired base saving the remainder as one of the new digits, then repeating the process with the new quotient until the quotient becomes zero. Let's examine this in an example. Convert 247D to binary. I'll use the notation,

```
dividend÷divisor=
  quotient°remainder
```

```
247 ÷ 2 = 123 ° 1
123 ÷ 2 = 61 ° 1
61 ÷ 2 = 30 ° 1
30 ÷ 2 = 15 ° 0
15 ÷ 2 = 7 ° 1
7 ÷ 2 = 3 ° 1
3 ÷ 2 = 1 ° 1
1 ÷ 2 = 0 ° 1
```

Taking the remainder digits in reverse sequence, we have 247D equals 11110111B. Another way would be to subtract off powers of two starting from the highest power of two not exceeding the value of the number and showing a "1" in the position representing each power of two subtracted.

An understanding of the binary number system is basic to understanding the concepts which follow. Therefore, if you are confused by any of this, spend some time with pencil and paper to practice some conversions to and from decimal.

Before I get to far astray, let me return to the subject of calculation. Now you have all heard the statement that computers are dumb. It's quite true. We may think that computers can do all sorts of things. In BASIC, all you have to do to tell the computer to calculate the square root of 9876 is write `PRINT SQR(9876)`. But computers really only know how to ADD. All other mathematical operations are built up from the lowly ADD operation! Because of this, its important for us to get a firm grasp in adding before I go on to more complicated operations.

But computers really only know how to ADD

Everyone knows how to add, don't they? Probably so. But I suspect that a lot of folks don't understand some of the finer points of adding because they don't focus on the minutia. But I have to focus on those finer points of the addition process to clarify binary addition operations. So I'll begin my discussion on adding by a review of decimal addition without omitting any minute details.

Addition is a positional operation. That's very important to understand. If we want to add up two numbers, we have to align their positional coefficients before we add up the digits. Taking as an example, the addition of 113 and 82, we are adding:

```
[1*100 + 1*10 + 3*1] + [8 *10 + 2*1] =
[1*100] + [1*10 + 8*10] + [3*1 + 2*1] =
[100] + [90] + [5] = 195
```

I aligned the two numbers so that the coefficients of like positions were added. The obvious way, and less confusing notationally, is to just align the numbers vertically and add the columns in our head. This produces:

```
 113
+82
===
195
```

That was an easy example; there was no "overflow" in any one column (position). What if there were? Say we wanted to add up 113 and 28? Aligning the numbers vertically and adding up the columns, I calculate:

```
 113
+28
===
141
```

Well how in the world did I get a "4" as the result of adding the second column? What happened was that when I added up the "3" and the "8" of the first column, the result was a value which *overflowed* the column position. Adding 8 + 3 yields 11, but 11 is too big a number to "fit" into that place. If you go back to early education addition, you may recollect that 8+3 does not equal 11; it equals "**1 carry the 1**"!

8+3 = 1 carry the 1

Thus in this example, we mentally say to ourselves while we add from right to left: eight plus three equals one *carry the one*, two plus one equals three plus the carry equals four, and one plus zero equals one. Experienced adders have zoomed right through these minute details for so many years that they have forgotten them. Notice that term, *carry*? That's an important term which indicates that the positional sum overflowed the storage capacity of the position.

Let's now switch to binary and see the same addition. Binary has a small addition table; the smallest of any number base. First look at positional binary column addition. It's table is:

```
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0C
```

Where a carry results from a column addition, I have noted that with the letter "C".

The binary representation of 113D is 01110001B and the binary representation of 28D is 00011100B. Here I have used an eight-bit representation, i.e. I have represented the number using eight binary digits. *Bit*, by the way, is a recognized shorthand way of saying "binary digit". We align the positions and add columns keeping track of overflow (carry), accumulating the carry into the immediately left-adjacent position.

```
 01110001
+00011100
=========
 10001101
```

Adding right to left: 1+0=1, 0+0=0, 0+1=1, 0+1=1, 1+1=0C, 1+0+C=0C, 1+0+C=0C, 0+0+C=1. One of the finer points about adding and overflow is that we never can have a double carry from a single position addition regardless of the number system base. Similarly, if we want to construct a mechanical adder, it needs only three inputs: one for the augend, one for the addend, and one for the carry in. It also has two outputs: the sum and the carry out.

I cannot overemphasize the necessity of understanding this simple binary addition. Spend some time in practicing it with other numbers before you tackle the next topic.

How can I subtract when all I can do is ADD?

One of the assumptions I must now make is that the reader understands the concept of negativity. No, that has nothing to do with the theory of relativity, so you non-physics folks have no excuse. Is there any distinction in calculating the difference [5 - 3] versus calculating the summation [5 + (-3)]? The result is the same; but the technique may be different.

Let's think of a world where there is no operation called subtraction. In this obscure world, there is only addition. This world does have negative numbers, but it doesn't have a minus sign. This world, which happens to use a decimal number system, has never had to count higher than 499; thus they represent negative values by the numbers 500-999. Now in any number system, it is important that the addition of two numbers having equal magnitude but opposite sense (sign) result in a zero value. Therefore, in this special world without subtraction, the sum of negative three and positive three must equal zero. They represent a positive three by the sequence, "003", therefore, a zero result would require negative three to be represented by "997". Why? Because 003 + 997 = 000 with a carry out of the hundreds place. That means negative one is represented by "999", and negative five hundred by "500". Note that the range of values represented by this system of numbers is negative five hundred to positive four ninety nine (or -500 to 499 in our normal world).

The number system just proposed uses what is called "*tens complement*" to represent negative numbers. A mathematician, or other ordinary person in our world, constructs the tens complement by adding one to the *nines complement*. That brings you to ask, what in the world is the nines complement? Good question. The nines complement is easy to present; simpler than the tens complement. To figure the nines complement of a decimal number, simply subtract each digit from nine. So the nines complement of 267 is 732; the tens complement would be 733. You see, therefore, that we can't represent a negative number by the nines complement because the sum of a number and its nines complement is all nines.

So why discuss the nines and tens complements? Because that should help you to understand the binary computer world which has only addition. In a fashion corresponding to the decimal system, a binary number has a *ones complement*. The ones complement of a binary number correlates to the nines complement of a decimal number. You just mentally subtract each binary digit from 1 to get the ones complement. But since "1-0=1" and "1-1=0", all you need to remember to take the ones complement of a binary number is to invert each bit. Change one bits to zero, and zero bits to one. That is the operation of *complementing*. In fact, when we use the term *complement*, we always imply ***ones complement.***

The binary system also has a complement which correlates to the tens complement of decimal; it is called the *twos complement*. The twos complement is simply the ones complement plus one. Most computers use a twos complement representation for indicating negative numbers. That's because as in the decimal case, 11111111 (negative one in twos complement) added to 00000001 (positive one) equals zero. To construct negative one in twos complement representation, start with positive one, complement it, then add one. Here's those steps in tabular form.

```
1. 00000001      positive one
2. 11111110      complemented
3. 11111111      one added
```

If we confine ourselves to a "signed" 8-bit binary number, its range, therefore, extends from -128D to +127D. We also can determine whether a binary number is representing a positive or a negative value by examining the most significant bit. If it is a "1", the number is negative; if a "0", the number is positive. We commonly call that bit position the *sign bit*. So now how do we subtract when we have no subtraction operation? We simply take the twos complement of the subtrahend and add it to the minuend. As easy as pie.

Well I think this is enough heavy stuff for one issue. I will pick up where I left off this topic in the next issue of TMQ.

Applications for the User

The following programs, although shown in their respective source code language, are nevertheless applications which may be directly usable by the non-programming user. All you need do is obtain the assembled/compiled program from the DISK NOTES 3.1 diskette which is associated with this issue of THE MISOSYS QUARTERLY.

Reading NEWDOS80 2.0 Disks From LDOS and LS-DOS

David Goben
67 Highland Rd.
Mansfield Ctr, CT 06250

If you have upgraded your system to LDOS or LS-DOS from NEWDOS80 2.0, you have been stuck with the chore of transferring your files from the NEWDOS80 environment to that of LDOS. Because of the difference in disk storage format, this normally involved saving the NEWDOS data to a single-density disk and then copying it from that disk to the LDOS disk. To ease this burden and remove that extra step, I have written a NEWDOS80 disk filter called NEWDOS/CMD. This program will operate under LDOS on the Model I and Model III, and under LS-DOS 6 on the Model 4. I wrote it in such a way that NEWDOS/CMD will recognize the environment that it is working under and to operate accordingly. For a reference on writing such transportable programs, see my article in the August 1987 issue of 80 MICRO (p. 86, 'A Disassembler for all DOSes'). Read the section labeled 'Making Programs Transportable' on page 88.

The conflict between the disk formats is that even though the sector naming is the same, the way that the operating systems handle them are vastly different. NEWDOS operates in such a way that it 'thinks' that the disk, even though it is double density, has only 10 sectors to a track and 5 sectors to a granule. What NEWDOS/CMD does is fool LDOS into thinking the same thing, and operate on it accordingly. Of course the actual communication with the disk must still be for 18-sector cylinders, and so it maintains its own drive code table within the filter to do the actual disk reads. However, the low memory DCT that the system THINKS it's using is set up for single-density disks, thus granule and cylinder calculations done by LDOS are computed properly for NEWDOS-type usage. What the filter does is take this calculated information, saves it, and recalculates the sector location pointers to the ACTUAL cylinder and relative sector.

NEWDOS/CMD is confined to single-sided, double density disks. It is also capable of reading either Model I or Model III NEWDOS80 version 2.0 disks (I do not know if those Model I owners who have their NEWDOS80 1.0 systems patched for double-density can use this). It will also work with PERCOM's DBLDOS disks.

To use NEWDOS/CMD, type the program listing into BASIC on your I/III/4 and RUN it. It will create a program called NEWDOS/CMD. This program you must use from DOS. The syntax for usage is

```
NEWDOS :d (1|3|Y)
```

where :d is the desired drive number, which must NOT be :0. 1 indicates Model I NEWDOS80 format. 3 represents the Model III version. Y will yank (remove) the filter from the drive and memory. If you wish to change between disk formats, you must first yank the previous filter, unless you are using a different drive. Notice that while the filter is installed that you can only use the specified disk format in it. Also note that you can only READ from the disk and not WRITE to it.

NEWDOS/CMD is based upon a program concept created by Roy Soltoff in Issue II of Notes from MISOSYS (p. 59). The program listed there was written for the Model I and III versions of LDOS. I took it further by streamlining the initialization and adding the capability to also operate under LS-DOS. Enjoy.

```
10 'NEWDOS/CMD & NEWDOS/BAS WRITTEN 1988 BY DAVID
GOBEN
20 CLS:PRINT"BUILDING 'NEWDOS/CMD'":RESTORE
30 OPEN"O",1,"NEWDOS/CMD":
L=90:HX$="0123456789ABCDEF"
40 CS=0:L=L+10
50 READ A$:IF A$="END"THEN CLOSE:END ELSE IF
LEFT$(A$,1)="-"THEN 80
```

```
60 A=INSTR(HX$,LEFT$(A$,1))*16 +
INSTR(HX$,RIGHT$(A$,1))-17
70 PRINT#1,CHR$(A);:CS=CS+A:GOTO 50
80 IF VAL(MID$(A$,2))=CS THEN 40 ELSE PRINT"CHECKSUM
ERROR IN LINE"L:END
90 ''' DATA AREA '''
100 DATA
05,09,00,00,00,4E,45,57,44,4F,53,05,09,00,00,-492
110 DATA
52,20,20,20,20,20,20,01,FE,00,52,CD,72,54,CD,-1219
120 DATA
C0,54,4E,45,57,44,4F,53,2F,38,30,20,32,2E,30,-1067
130 DATA
20,44,49,53,4B,20,46,49,4C,54,45,52,20,46,4F,-998
140 DATA
52,20,4D,4F,44,45,4C,20,31,2F,33,2F,34,20,4C,-869
150 DATA
44,4F,53,2F,4C,53,2D,44,4F,53,2E,0D,43,4F,50,-996
160 DATA
59,52,49,47,48,54,20,28,43,29,20,31,39,38,38,-901
170 DATA
20,42,59,20,44,41,56,49,44,20,47,4F,42,45,4E,-974
180 DATA
2E,20,41,4C,4C,20,52,49,47,48,54,53,20,52,45,-975
190 DATA
53,45,52,56,45,44,0D,52,45,4C,45,41,53,45,44,-1051
200 DATA
20,49,4E,54,4F,20,54,48,45,20,50,55,42,4C,49,-1015
210 DATA
43,20,44,4F,4D,41,49,4E,20,46,4F,52,20,46,52,-986
220 DATA
45,45,20,44,49,53,54,52,49,42,55,54,49,4F,4E,-1098
230 DATA
2E,0D,0D,00,7E,23,FE,20,28,FA,FE,3A,28,F6,FE,-1661
240 DATA
0D,20,1D,CD,C0,54,50,41,52,41,4D,45,54,45,52,-1228
250 DATA
20,45,52,52,4F,52,0D,00,CD,8E,54,C2,2D,40,3E,-1235
260 DATA
16,EF,32,BD,54,D6,30,38,DC,FE,08,30,D8,B7,28,-1871
270 DATA
D5,4F,CD,DA,54,FD,7E,00,FE,C9,20,12,CD,B3,54,-2151
280 DATA
4E,4F,54,20,41,43,54,49,01,FE,FC,52,56,45,0D,-1319
290 DATA
00,C3,CF,52,7E,23,FE,20,28,FA,FE,28,C2,BB,52,-1978
300 DATA
46,23,7E,FE,0D,28,05,FE,29,C2,BB,52,78,FE,59,-1764
310 DATA
CA,D1,53,FE,79,CA,D1,53,FE,31,28,08,D6,33,C2,-2173
320 DATA
BB,52,32,43,55,CD,9A,54,20,2C,CD,C0,54,46,49,-1614
330 DATA
4C,54,45,52,20,41,4C,52,45,41,44,59,20,49,4E,-1040
340 DATA
53,41,54,4C,4C,45,44,00,CD,C0,54,20,4F,4E,20,-1223
350 DATA
00,CD,B3,54,0D,00,C3,CF,52,FD,E5,E1,11,4E,55,-1852
360 DATA
01,0A,00,ED,B0,DD,21,4E,55,DD,7E,03,E6,07,F6,-1674
370 DATA
C0,DD,77,03,DD,7E,04,E6,0F,F6,40,DD,77,04,CD,-1990
380 DATA
E4,54,22,13,55,E5,1B,D5,A7,ED,52,EB,2A,28,55,-1807
390 DATA
19,22,28,55,2A,48,55,19,22,48,55,E1,D1,01,47,-1105
400 DATA
00,ED,B8,EB,CD,FD,54,23,FD,74,02,FD,75,01,FD,-2228
410 DATA
36,07,09,FD,36,08,24,CD,C0,54,46,49,4C,54,45,-1274
420 DATA
52,20,49,4E,53,54,41,4C,4C,45,44,00,C3,52,53,-1146
430 DATA
CD,9A,54,28,19,CD,C0,54,46,49,4C,54,45,52,20,-1475
440 DATA
4E,4F,54,20,50,52,45,53,45,4E,54,00,C3,52,53,-1178
450 DATA
FD,66,02,FD,6E,01,E5,01,3D,01,FE,F8,53,00,09,-1607
460 DATA
E5,DD,E1,DD,46,02,DD,4E,01,FD,70,02,FD,71,01,-2002
470 DATA
CD,E4,54,23,D1,A7,ED,52,28,3B,CD,C0,54,46,49,-1970
480 DATA
4C,54,45,52,20,59,41,4E,4B,45,44,2C,20,42,55,-1014
490 DATA
54,20,43,41,4E,4E,4F,54,20,52,45,43,4C,41,49,-1031
500 DATA
4D,20,48,49,47,48,20,4D,45,4D,4F,52,59,20,53,-1017
510 DATA
50,41,43,45,0D,00,C3,CF,52,EB,23,23,5E,23,56,-1298
520 DATA
EB,CD,FD,54,CD,C0,54,46,49,4C,54,45,52,20,59,-1833
530 DATA
41,4E,4B,45,44,20,46,52,4F,4D,20,00,C3,5A,53,-1095
540 DATA
E5,21,00,00,7E,47,2F,77,BE,70,28,0E,3A,25,01,-1077
550 DATA
FE,49,3E,02,28,01,AF,3C,32,91,54,E1,C9,C5,47,-1640
560 DATA
3E,00,B7,28,02,FE,02,78,C1,C9,FD,66,02,FD,6E,-1777
570 DATA
01,23,23,23,23,11,15,55,1A,BE,C0,47,13,23,1A,-823
580 DATA
BE,C0,10,F9,C9,CD,C0,54,44,52,49,56,45,20,3A,-1797
590 DATA
30,20,00,E3,7E,23,B7,28,05,CD,CD,54,18,F6,E3,-1687
600 DATA
C9,CD,8E,54,C2,33,00,C5,4F,3E,02,EF,C1,C9,CD,-2055
610 DATA
8E,54,C2,8F,47,3E,51,EF,C9,CD,8E,54,28,0A,38,-1754
620 DATA
04,2A,11,44,C9,2A,49,40,C9,C5,01,5C,F4,54,21,-1363
630 DATA
00,00,44,3E,64,EF,C1,C9,CD,8E,54,28,0A,38,04,-1404
640 DATA
22,11,44,C9,22,49,40,C9,C5,06,00,18,E7,18,0A,-1184
650 DATA
00,00,07,24,4E,45,57,44,4F,53,78,FE,0B,3E,0F,-969
660 DATA
D0,FD,E5,D5,FD,21,4E,55,E5,D5,6A,26,00,54,5D,-2115
670 DATA
29,29,19,29,D1,16,00,19,1E,12,AF,3C,ED,52,30,-1054
680 DATA
FB,19,3D,3C,57,5D,E1,CD,4E,55,D1,FD,E1,C9,02,-2060
690 DATA 02,00,52,-84,END
```

For those interested in the assembler source code for this program, the following is the assembler source.

```
;-------------------------------------------------
;NEWDOS/80 2.0 DISK FILTER
; FOR MODEL 1/3/4 LDOS/LS-DOS.
;
;COPYRIGHT (C) 1988 BY DAVID GOBEN.
; ALL RIGHTS RESERVED
; RELEASED INTO THE PUBLIC
; DOMAIN FOR FREE DISTRIBUTION.
;-------------------------------------------------
;USAGE:
```

```
;       NEWDOS :d (1|3|Y)
;WHERE:
; :d is the disk to use. ':' is optional.
; Drive :0 is illegal.
; parameter 1 = Model I Double
; Density NEWDOS/80 2.0 Disk.
; parameter 3 = Model III
; NEWDOS/80 2.0 disk
; parameter Y = YANK (remove)
; Filter from system
;*********************************
;NOTE ON IMPLEMENTATION: To change
; to ALDS format, delete the SVC
; MACRO defined below, change the
; ORG to PSECT, and change all DB's
;in front of message lines to DEFM.
;***********************************************
CR      EQU     13      ;carriage return

SVC     MACRO   OPR
        LD      A,OPR
        RST     28H
        ENDM
;-----------------------------------------------
;                       start of program
        ORG     5200H
;set computer model
MAIN    CALL    SETMDL
;print a message
        CALL    PRINT
        DB 'NEWDOS/80 2.0 DISK FILTER FOR MODEL
1/3/4 LDOS/LS-DOS.'
        DB      CR
        DB      'COPYRIGHT (C) 1988 BY DAVID
GOBEN. ALL RIGHTS RESERVED'
        DB      CR
        DB      'RELEASED INTO THE PUBLIC DOMAIN
FOR FREE DISTRIBUTION.'
        DB      CR,CR,0
;skip over spaces and drive leader
CHK     LD      A,(HL)
        INC     HL
        CP      ' '
        JR      Z,CHK
        CP      ':'
        JR      Z,CHK
;carriage return?
        CP      CR
        JR      NZ,MAIN2
PERR    CALL    PRINT
        DB      'PARAMETER ERROR'
        DB      CR,0
EXIT    CALL    MODEL
        JP      NZ,402DH        ;exit to dos on
1/3
        SVC     22              ;exit on 4

MAIN2   LD      (MSG),A ;save code
        SUB     '0'             ;check
        JR      C,PERR  ; range
        CP      8
        JR      NC,PERR
        OR      A
        JR      Z,PERR
;set drive number to C
        LD      C,A
;get its drive code table
        CALL    GETDCT
        LD      A,(IY+0)        ;active?
        CP      0C9H
        JR      NZ,MAIN3        ;yes
;display drive message
        CALL    DRIVE
        DB      'NOT ACTIVE'
        DB      CR,0
        JP      EXIT

MAIN3   LD      A,(HL)          ;find '('
        INC     HL
        CP      ' '             ;skip spaces
        JR      Z,MAIN3
        CP      '('
        JP      NZ,PERR
        LD      B,(HL)          ;save parameter
        INC     HL              ;find end
        LD      A,(HL)          ; of data
        CP      CR
        JR      Z,MAIN4
        CP      ')'
        JP      NZ,PERR
MAIN4   LD      A,B             ;get parm
        CP      'Y'             ;yank?
        JP      Z,YANK
        CP      'y'
        JP      Z,YANK
        CP      '1'             ;Model 1?
        JR      Z,MAIN5
        SUB     '3'
        JP      NZ,PERR
        LD      (BUMP),A        ;turn off INC A
;see if driver already installed
MAIN5   CALL    CKDRVR
        JR      NZ,MAIN6        ;not, so OK
        CALL    PRINT
        DB 'FILTER ALREADY INSTALLED'
        DB      0
DSPON   CALL    PRINT
        DB      ' ON '
        DB      0
DSPDRV  CALL    DRIVE
        DB      CR,0
        JP      EXIT
MAIN6   PUSH    IY              ;set DCT to HL
        POP     HL
        LD      DE,LAST         ;append DCT
        LD      BC,10           ; to filter
        LDIR
        LD      IX,LAST         ;Set up filter's
        LD      A,(IX+3)        ; DCT
;save time delay and step rate
        AND     7
;set WP bit and indicate DDEN
        OR      0C0H
        LD      (IX+3),A
        LD      A,(IX+4)        ;save drive
        AND     0FH             ; select
        OR      40H             ;set DDEN
        LD      (IX+4),A        ; bit
        CALL    GETMEM          ;get high$
        LD      (NEWDOS+2),HL   ;into filter
        PUSH    HL              ;save value
;set to top of program
        DEC     DE
        PUSH    DE              ;save addr
;reset carry flag
        AND     A               ;compute
        SBC     HL,DE           ; offset
        EX      DE,HL           ;to DE

        LD      HL,(REL1)       ;apply offset to
        ADD     HL,DE           ; filter
        LD      (REL1),HL       ; addresses
        LD      HL,(REL2)
        ADD     HL,DE
```

```
        LD      (REL2),HL
;get top of filter
        POP     HL
        POP     DE              ;get himem
;set length of file
        LD      BC,LAST-NEWDOS+10
;move it to high memory
        LDDR
        EX      DE,HL           ;set new
        CALL    SETMEM   ; himem
        INC     HL              ;set driver
        LD      (IY+2),H        ; address
        LD      (IY+1),L        ; into DCT
;set sector highname (fake it)
        LD      (IY+7),9
;set fake allocation
        LD      (IY+8),24H
        CALL    PRINT
        DB      'FILTER INSTALLED'
        DB      0
        JP      DSPON
;---------------------------------
;YANK FILTER FROM DRIVE
YANK    CALL    CKDRVR          ;driver
        JR      Z,YANK2         ; present?
        CALL    PRINT
        DB      'FILTER NOT PRESENT'
        DB      0
        JP      DSPON
YANK2   LD      H,(IY+2)        ;get driver
        LD      L,(IY+1)        ; address
        PUSH    HL              ;save copy
;point HL to end of filter
        LD      BC,LAST-NEWDOS
        ADD     HL,BC
        LD      IX,HL           ;into IX
        LD      B,(IX+2)        ;driver to
        LD      C,(IX+1)        ; reg BC
        LD      (IY+2),B        ;put it to
        LD      (IY+1),C        ; DCT
        CALL    GETMEM          ;get himem
;point (hopefully) to current driver
        INC     HL
        POP     DE
        AND     A
        SBC     HL,DE           ;same?
        JR      Z,YANK3         ;yes
        CALL    PRINT
        DB      'FILTER YANKED, BUT CANNOT RECLAIM
HIGH MEMORY SPACE'
        DB      CR,0
        JP      EXIT
;filter address to DE
YANK3   EX      DE,HL
;point to old himem
        INC     HL
        INC     HL
        LD      DE,(HL)         ;get it
        EX      DE,HL           ;reset
        CALL    SETMEM          ; himem
        CALL    PRINT
        DB      'FILTER YANKED FROM '
        DB      0
        JP      DSPDRV
;************************************************
;               SUBROUTINES
;************************************************
;figure which computer we're working with
;save buffer pointer
SETMDL  PUSH    HL
        LD      HL,0            ;point to lowmem
        LD      A,(HL)
        LD      B,A             ;save value
        CPL                     ;change it
        LD      (HL),A          ;set new
        CP      (HL)            ;Z if RAM
        LD      (HL),B          ;old value
;is ram, so ignore more tests
        JR      Z,MDLX
;else test for Model III
        LD      A,(125H)
        CP      'I'
        LD      A,3-1           ;init for
        JR      Z,$+3           ; Model 3
        XOR     A               ;init for
        INC     A               ; Model 1
        LD      (MOD+1),A       ;set flag
MDLX    POP     HL              ;get
        RET                     ; pointer
;-------
;set Z for Model 4,
; NZ/NC for Model III,
; and NZ,C for Model I
MODEL   PUSH    BC              ;save reg
        LD      B,A             ;save regA
MOD     LD      A,0             ;Model type
        OR      A               ;set Z for
        JR      Z,$+4           ; Model 4
        CP      2
        LD      A,B
        POP     BC
        RET
;-------
;see if driver present
CKDRVR  LD      H,(IY+2)        ;get address
        LD      L,(IY+1)        ; from DCT
        INC     HL              ;point to filter
        INC     HL              ; name
        INC     HL
        INC     HL
        LD      DE,NAME         ;current filter
        LD      A,(DE)          ; name
        CP      (HL)            ;match?
        RET     NZ              ;no
        LD      B,A             ;else set count
CKLP    INC     DE              ;point to a byte
        INC     HL
        LD      A,(DE)          ;name byte
        CP      (HL)            ; match?
        RET     NZ              ;no
        DJNZ    CKLP            ;check next
        RET
;-------
;display drive
DRIVE   CALL    PRINT
        DB      'DRIVE :'
MSG     DB      '0 '
        DB      0
;-------
;display a message
PRINT   EX      (SP),HL
        LD      A,(HL)          ;get a byte
        INC     HL              ;bump pointer
        OR      A               ;done?
        JR      Z,PRINTX
        CALL    DSP             ;display it
        JR      PRINT+1         ;and go for
PRINTX  EX      (SP),HL         ; another
        RET
;-------
;display a byte
DSP     CALL    MODEL           ;display on
        JP      NZ,33H          ; Model I/III
        PUSH    BC              ;save used
```

```
        LD      C,A             ; register
        SVC     2
        POP     BC
        RET
;-------
;get DCT for drive C
GETDCT  CALL    MODEL
        JP      NZ,478FH
        SVC     81
        RET
;-------
;get high memory
GETMEM  CALL    MODEL
        JR      Z,GET4
        JR      C,GET1
        LD      HL,(4411H)
        RET
GET1    LD      HL,(4049H)
        RET
GET4    PUSH    BC
        LD      HL,$-$
        LD      B,H
GET4B   SVC     100
        POP     BC
        RET
;-------
;put high memory
SETMEM  CALL    MODEL
        JR      Z,SET4
        JR      C,SET1
        LD      (4411H),HL
        RET
SET1    LD      (4049H),HL
        RET
SET4    PUSH    BC
        LD      B,0
        JR      GET4B
;************************************************
;               NEWDOS/80 FILTER ROUTINE
;************************************************
NEWDOS  JR      FILTER
        DW      $-$             ;high mem save
NAME    DB      7
        DB      '$NEWDOS'
FILTER  LD      A,B             ;get function
        CP      0BH
;set Write Protected Disk error
        LD      A,0FH
        RET     NC              ;not allowed
        PUSH    IY              ;else save regs
        PUSH    DE
        LD      IY,LAST         ;pt to new DCT
REL1    EQU     $-2
        PUSH    HL
        PUSH    DE
        LD      L,D             ;set cyl to HL
        LD      H,0
        LD      D,H             ;set DE to same
        LD      E,L
        ADD     HL,HL           ;*2
        ADD     HL,HL           ;*4
        ADD     HL,DE           ;*5
        ADD     HL,HL           ;*10
        POP     DE              ;get cyl/sec
        LD      D,0             ;DE = sectors
;HL = total sector count
        ADD     HL,DE           ;now divide
        LD      E,18            ; it by 18
        XOR     A               ;init counter
        INC     A
        SBC     HL,DE
        JR      NC,$-3
        ADD     HL,DE
        DEC     A               ;set cyls to A
BUMP    INC     A               ;adj for Model I
        LD      D,A             ;set cyl
        LD      E,L             ;set sector
        POP     HL
        CALL    LAST            ;Do function
REL2    EQU     $-2
        POP     DE              ;get regs back
        POP     IY
        RET
LAST    EQU     $
;************************************************
        END     MAIN
```

XMODEM/CCC
by David B. Lamkins

Fm MISOSYS, Inc: David wrote me a letter "complaining" about MC's automatic expansion of the tab character to multiple spaces. Luckily, I had him on that one as that particular operation is covered in the MC manual. Of course, an index to the manual would have provided the ability to find the reference. In any event, after informing David that the ioctl() function could be used to alter tab expansion, David sent me the following. See what happens when I'm right? I get a TMQ article! Now with David's work, there's no excuse for not having an XMODEM file transfer utility.

An XMODEM program is used to facilitate the transfer of files from one computer system to another through a communications channel - typically the serial port of a TRS-80. With both the transmitting and receiving ends equipped with an XMODEM program, a file can be sent with a modest amount of error checking. This is one means of transferring files to/from an MS-DOS based computer.

Fm David B. Lamkins: Thank you for your response to my problems in using the ioctl() function from MC. I went back and tried it again and was able to get it to work as expected. I suspect that my original problem was due to lack of diligence and/or sleep. Next time, I won't be so quick to send a letter...

In apology for overlooking the obvious, I've enclosed a diskette containing a simple XMODEM program written for MC. The program has been tested on the Model III under LDOS 5.3, and does indeed demonstrate the proper use of the ioctl() function in a communications' application. You may, if you wish, include the code in either *THE MISOSYS QUARTERLY* or Disk NOTES. [-ed note: I compiled versions for both LDOS and LS-DOS from David's source which follows. The program works equally well on both machines (Model III and Model 4). Both versions, as well as the source file, are on DISK NOTES 3.1].

David B. Lamkins
25 Sutcliffe Ave.
Canton, MA 02021

```
/*   xmodem/ccc

     Written by David B. Lamkins, June 1988

     Source compiler: MISOSYS MC
     Target computer: TRS-80 Model III

     This program implements the XMODEM protocol for
     transmission and reception of files between the
     local computer and a remote system running
XMODEM.
     The CRC variant of XMODEM is not implemented.

     The RS232T driver must be installed for the *CL
     device, using a command similar to
          SET *CL RS232T (W=8,P=N,B=1200

     XMODEM assumes an 8-bit communication channel,
     therefore you MUST be certain to configure the
     *CL driver for 8-bit data.

     The program is invoked using a command like
          XMODEM *CL -S SENDFILE/TXT
     to send a file, or like
          XMODEM *CL -R RECVFILE/TXT
     to receive a file.  During operation, a '*' will
     be displayed for each successful block and a '?'
     for each unsucessful block transferred.

     This program may work, but has not been tested,
      on other machines under other compilers.
*/

#include <stdio.h>
#include <sgtty.h>

#ifdef MC
#option REDIRECT OFF
#option ERRORMSG OFF
#endif

#define como(ch) fputc(ch,cl)
#define comi() fgetc(cl)
#define comc() checkc(cl)
#define akey() checkc(stdin)

#define NAK 0x15        /* negative acknowledge */
#define ACK 0x06        /* acknowledge: block received
OK */
#define SOH 0x01        /* block header */
#define EOT 0x04        /* no more blocks to send */
#define CAN 0x18        /* cancel transmission */
#define NOTIMEOUT 0
#define TIMECONST 10    /* transaction timeout */
#define SHORTTIME 1         /* character timeout */
#define XBLKSIZE 128    /* data block length */
#define MODULUS 256         /* cardinality of data */
#define MODMASK (MODULUS-1)
#define TRYLIMIT 10         /* retry limit */

#define CHECKOK 0
#define CHECKBAD 1
#define CHECKABORT -1
#define TIMEDOUT -1
#define ACKSAME -1

#define BANNER "XMODEM by David B. Lamkins, June
1988"
#define USAGE "usage: xmodem <channel> -<s|r>
<file>"
#define NOFILE "can't open file"
#define NOCHAN "can't open comm channel"
#define IOCERR "can't IOCTL comm channel"
#define NOWRITE "can't write output file"
#define STARTED "started..."
#define INTERRUPTED "aborted"
#define SUCCESS "completed"
#define GIVEUP "failed"

FILE *cl, *fo, *fi;

/*****************/
/* Mainline code */
/*****************/

main(argc, argv)
int argc; char **argv;
{
     static struct sgttyb sg = {0,0};

     puts(BANNER);
     if (argc != 4) leave(USAGE);
     if (!(cl = fopen(*++argv, "r+"))) leave(NOCHAN);
     if (ioctl(fileno(cl), TIOCGETP, &sg))
leave(IOCERR);
     sg.sg_control &= ~(IO_TABSTOP|IO_BREAK);
     if (ioctl(fileno(cl), TIOCSETP, &sg))
leave(IOCERR);
     if (**++argv != '-') leave(USAGE);
     switch (*++*argv) {
     case 'r': case 'R': dorecv(*++argv); break;
     case 's': case 'S': dosend(*++argv); break;
     default: leave(USAGE);
     }
}

leave(s)
char *s;
{
     putchar('\n');
     puts(s);
     exit();
}

chatter(blk, flg)
int blk, flg;
{
     static int col = 0;
     char cvbuf[33];

     putchar(flg == CHECKOK ? '*' : '?');
     if (col++ == 32) {
          puts(itoa(blk,cvbuf));
          col = 0;
     }
}

/****************/
/* Receive code */
/****************/

dorecv(s)
char *s;
{
```

```
    int blknum, tries;
    char *data;

    if (!(fo = fopen(s, "w"))) leave(NOFILE);

    como(NAK);
    puts(STARTED);
    blknum = 1;
    tries = TRYLIMIT;
    while (TRUE) {
        if (data = recvblock(blknum % 256)) {
            if (data != ACKSAME) {
                chatter(blknum, CHECKOK);
                putblock(data);
                ++blknum;
            }
            como(ACK);
            tries = 10;
        } else {
            chatter(blknum, CHECKBAD);
            if (--tries == 0) {
                como(CAN);
                leave(GIVEUP);
            }
            como(NAK);
        }
    }
}

putblock(data)
char *data;
{
    if (!fwrite(data, 1, XBLKSIZE, fo))
        leave(NOWRITE);
}

char *recvblock(blknum)
int blknum;
{
    static char data[XBLKSIZE];
    int i, blk, blkc, ch, csum;

    if (rwaitfor(SOH, TIMECONST) == CHECKBAD) return
NULL;
    if ((blk = timedrecv(SHORTTIME)) == TIMEDOUT)
return NULL;
    if ((blkc = timedrecv(SHORTTIME)) == TIMEDOUT)
return NULL;
    if (blk != (~blkc & MODMASK)) return NULL;
    if (blk == (blknum-1) % MODULUS) return ACKSAME;
    if (blk != blknum) return NULL;
    for (i = 0, csum = 0; i < XBLKSIZE; ++i) {
        ch=timedrecv(SHORTTIME);
        if (ch == TIMEDOUT) return NULL;
        csum += data[i] = ch;
    }
    return ((ch=timedrecv(SHORTTIME)) == TIMEDOUT ||
        (csum & MODMASK) != ch) ? NULL : data;
}

rwaitfor(c, secs)
char c;
int secs;
{
    int rxch;
    long expire;
    extern long time();

    expire = secs + time(NULL) + 1;
    while (!secs || time(NULL) < expire) {
        if (comc(cl))
            switch(rxch = comi(cl)) {
                case EOT: leave(SUCCESS);
                case CAN: leave(GIVEUP);
                default:  if (c == rxch) return
CHECKOK;
            }
        if (akey()) leave(INTERRUPTED);
    }
    return CHECKBAD;  /* timeout */
}

timedrecv(secs)
int secs;
{
    int rxch;
    long expire;
    extern long time();

    expire = secs + time(NULL) + 1;
    while (!secs || time(NULL) < expire)
        if (comc(cl)) return comi(cl);
    return TIMEDOUT;  /* timeout */
}

/*************/
/* Send code */
/*************/

dosend(s)
char *s;
{
    int tries, resp, blknum;
    char *data;

    if (!(fi = fopen(s, "r"))) leave(NOFILE);
    if (swaitfor(NAK, NOTIMEOUT) == CHECKABORT)
        leave(INTERRUPTED);
    puts(STARTED);
    blknum = 1;
    while (data = getblock()) {
        tries = TRYLIMIT;
        do {
            sendblock(data, blknum % MODULUS);
            resp = swaitfor(ACK, TIMECONST);
            chatter(blknum, resp);
        } while (resp == CHECKBAD && --tries);
        if (resp == CHECKABORT || !tries) break;
        ++blknum;
    }
    if (data && resp) {
        como(CAN);
        if (resp == CHECKBAD) leave(GIVEUP);
        else leave(INTERRUPTED);
    } else {
        como(EOT);
        leave(SUCCESS);
    }
}

char *getblock()
{
    static char block[XBLKSIZE];
    int i, c;

    fill(block, XBLKSIZE, NULL);
    for (i = 0; i < XBLKSIZE && (c = getc(fi)) !=
EOF; ++i)
        block[i] = c;
    return i == 0 ? NULL : block;
}

sendblock(data, blknum)
char *data; int blknum;
```

```
{
    int i, csum;

    como(SOH);
    como(blknum);
    como(~blknum & MODMASK);
    for (i = 0, csum = 0; i < XBLKSIZE; ++i) {
        como(data[i]);
        csum += data[i];
    }
    como(csum & MODMASK);
}

swaitfor(c, secs)
char c;
int secs;
{
    int rxch;
    long expire;
    extern long time();

    expire = secs + time(NULL) + 1;
    while (!secs || time(NULL) < expire) {
        if (checkc(cl))
            switch (rxch = getc(cl)) {
            case CAN: return CHECKABORT;
            case NAK: return CHECKBAD;
            default: if (c == rxch) return CHECKOK;
            }
        if (akey()) return CHECKABORT;
    }
    return CHECKBAD;  /* timeout */
}
```

LBMOVE/BAS
An LB Archival Utility
by Daniel L. Srebnick

Roy, First off, congrats on the birth of your son. Hope that Brenda and he are doing well. Here's the archival utility for Little Brother. LBMOVE/BAS is the ASCII program listing.

I hope that you and your readers find my little utility to be of help. I know that for me, it complements Little Brother quite nicely. Perhaps when you get around to the rewrite, you might want to consider offering this type of archival as a standard feature. I am sure that many users want a package that includes archival as a standard database function.

Daniel L. Srebnick
11 Brainard Avenue
Port Monmouth, New Jersey
07758

Database users often want to archive records which meet selected criteria. For example, a mail order firm maintains two different mailing lists. The first list contains names of likely prospects; say those who have ordered within the last eighteen months. The second list contains the names of customers who last ordered long ago. When a customer has not ordered within an eighteen month period, the company policy is to move that person from the first to the second list.

Other than manually deleting the record from the file for list 1 and then adding it to list 2, the solution to this problem is to have a means of automating the process. Initially, it seemed as if the MISOSYS LB Maintenance Utility would be able to perform this automation. The Maintenance Utility will write selected records from any database to a new file, but will not append records to an already existing file.

I set out to design a quick and dirty program in BASIC that would perform the following archival tasks:

(1) Read a Little Brother index which contains the record numbers from the input file that you wish to move to the output file. This must have been previously selected using the LB Sort and Select Records function.

(2) Write the selected records to an output file with matching file specifications.

(3) Delete the copied records from the original file.

In my efforts, I was limited by BASIC's constraint that the maximum number of records in a file may not exceed 32,767. This is because a record number in BASIC is represented in integer format and may not be a negative number. 32,767 is the largest positive integer that may be represented in signed integer format in the two bytes allowed. This limitation makes this program an excellent candidate for translation to C. In C, you can declare a variable as an unsigned integer, allowing for values up to 65535.

This program, therefore, will not support input or output files that are in excess of 32,767 records. Unless you are dealing with tremendous databases, this restriction will probably not matter too much. If the number of records that you have selected for archival will make the output file exceed 32,767 records, LBMOVE will move only as many records from the first file as can fit in the second, within the specified limit. It is also the responsibility of the user to make sure that the two files have the same structure before running LBMOVE. It is always advisable to have backups of the original files handy, especially if you are not sure about what you are doing.

A particular challenge prior to writing this program was to determine the exact structure of the Little Brother database and definition files. I did quite a bit of peeking around with a disk sector editor to find different data pointers and counters and got a couple of tips from Roy Soltoff and Joe Kyle-DiPietropaolo online via the CompuServe LDOS Forum. My findings are presented in table one along with documentation on the variables in my program that use the various values.

Disk space efficiency is always a consideration when dealing with a database application. Therefore, LBMOVE makes use of the deleted record chain in the output file and will reuse any record that Little Brother has marked as deleted.

One of the nice things that interpreted BASIC will do is to let you dimension an array based upon a variable subscript. This is necessary because the logical record length of a file is not know until its corresponding /DEF file is read. The record length is stored in integer variable LRL, which is then used in the OPEN statement. Since the file field buffer must also be dynamically declared, a loop is entered immediately after the OPEN fielding the input and output file buffers as arrays of one byte strings of DIMension(LRL). This is a good technique for dealing with file records of unknown length that only interpreted BASIC will allow. Incidentally, forget about compiling this program, the compiler will choke at line 140! The BASIC compilers do not take kindly to allowing dynamically dimensioned arrays with bounds set by run time input.

Long descriptive variable names are used throughout, with the file buffer equivalent of a program variable differing only in the $ suffix. The program will, of course, run faster if you delete the comments from your run time version. Subroutines begin at line 1000. If you delete the REM statement at the beginning of a subroutine, be sure to correct the associated GOSUB statement to call the following line.

BASIC should be entered with the (F=5) parameter, as LBMOVE uses five files. Lines 20 and 30 prompt you for the input and output filespecs. The input file is the name of the Little Brother database from which you wish to move records. Do not include the file extension. The output file is the name of the database to which you are moving records. No extension necessary here either. You are then prompted for the index number. This is the number of the previously created selection index for the input file. Make sure that all files are online before hitting enter.

Lines 50 - 200 open and field files and load the various pointers and counters, such as the allocated record count and the number of the last used record in the output file, the logical record length, and the pointer to the deleted record chain, so that deleted records are reused before any new records.

The main processing loop begins at line 220. INDEX.NUM hold the number of selected records that exist in the LB index file. IN.REC is the record number of the next selected input record to be moved. Line 260 makes a check for deleted records to use. A call is made to the appropriate record allocation subroutine at line 1000 or 2000 and in turn, these subroutines will call the write subroutine at line 3000. If all has gone well so far, the subroutine at 5000 will be called to delete the moved record from the input file. The main loop ends at line 270 and a message is printed indicating the number of records that were successfully moved. At this point, closing housekeeping is conducted to update the various altered record allocation pointers, the files are closed, and the program is terminated.

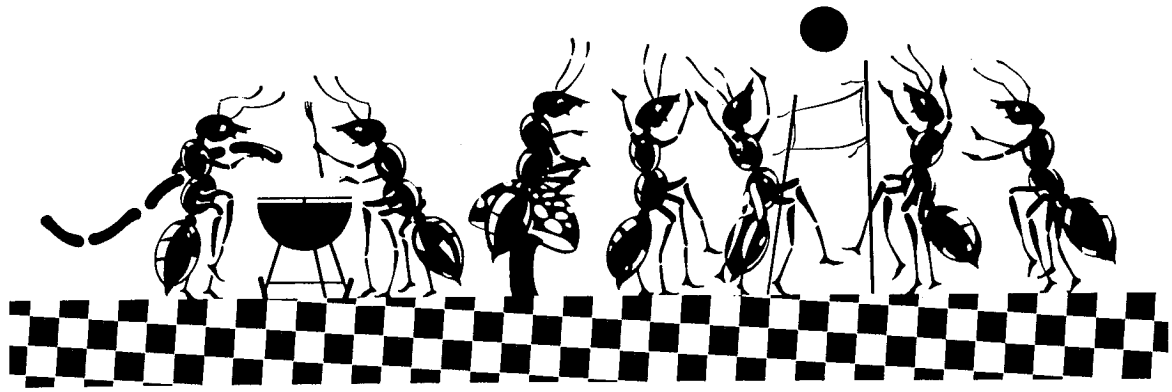
```
0 REM LBMOVE/BAS by Daniel L. Srebnick (c) 1988
                    PO Box 434
                    Port Monmouth, NJ   07758
                    CIS 72216,520
5 ON ERROR GOTO 6000
10 CLS
15 DEFINT I,A,L,D,S,P
20 INPUT "Input filespec";IN.FILE$
30 INPUT "Output filespec";OUT.FILE$
40 INPUT "Read which index (1-5)";INDEX.NUMBER
45 INDEX.NUMBER=INDEX.NUMBER-1
50 OPEN "R",1,IN.FILE$+"/sl"
   +RIGHT$(STR$(INDEX.NUMBER),1),2
60 FIELD 1,2 AS IN.REC$
70 OPEN "R",2,OUT.FILE$+"/def"
71 OPEN "R",5,IN.FILE$+"/def"
80 FIELD 2, 26 AS DUMMY2A$,2 AS ALLOC.REC$,
   2 AS LAST.REC$,2 AS LRL$,2 AS DEL.PTR$,
   55 AS DUMMY2B$,2 AS DEL.COUNT$,165 AS DUMMY2C$
81 FIELD 5,26 AS DUMMY5A$,2 AS IN.ALLOC.REC$,
   2 AS IN.LAST.REC$,2 AS IN.LRL$,2 AS IN.DEL.PTR$,
55 AS DUMMY5B$,2 AS IN.DEL.COUNT$,165 AS DUMMY5C$
90 GET 2
100 ALLOC.REC=CVI(ALLOC.REC$)
110 LAST.REC=CVI(LAST.REC$)
120 LRL=CVI(LRL$)
130 DEL.PTR=CVI(DEL.PTR$)
135 DEL.COUNT=CVI(DEL.COUNT$)
136 GET 5
137 IN.DEL.PTR=CVI(IN.DEL.PTR$)
138 IN.DEL.COUNT=CVI(IN.DEL.COUNT$)
140 IF LRL=256 THEN OPEN "R",3,IN.FILE$+"/lb"
    ELSE OPEN "R",3,IN.FILE$+"/lb",LRL
150 IF LRL=256 THEN OPEN "R",4,OUT.FILE$+"/lb"
    ELSE OPEN "R",4,OUT.FILE$+"/lb",LRL
151 IF LOF(3) > 32767 OR LOF(4) > 32767 THEN SOUND
    7,0:PRINT:PRINT "This program does not support
    files in excess of 32767 records.":CLOSE:END
160 DIM IN.BUFF$(LRL),OUT.BUFF$(LRL)
170 FOR SETUP=1 TO LRL
180 FIELD 3,SETUP-1 AS DUMMY$,1 AS IN.BUFF$(SETUP)
190 FIELD 4,SETUP-1 AS DUMMY$,1 AS OUT.BUFF$(SETUP)
200 NEXT SETUP
205 GET 1
210 INDEX.NUM=CVI(IN.REC$)
    'number of records to be moved
215 GET 1,3
    'position to prepare to get record numbers
217 PRINT:PRINT "Working..."
220 FOR LOOP=1 TO INDEX.NUM
    'go thru once for each record
230 GET 1 'get the next record number to process
240 IN.REC=CVI(IN.REC$)
250 GET 3,IN.REC+1 'read the selected record
260 IF NOT(DEL.PTR=0) THEN GOSUB 1000
    ELSE GOSUB 2000
265 GOSUB 5000 'subroutine to delete input record
270 NEXT LOOP
300 PRINT LOOP-1" records were copied from "
    IN.FILE$" to "OUT.FILE$"."
310 LSET LAST.REC$=MKI$(LAST.REC)
320 LSET DEL.PTR$=MKI$(DEL.PTR)
325 LSET DEL.COUNT$=MKI$(DEL.COUNT)
330 PUT 2,1:CLOSE 2
331 LSET IN.DEL.PTR$=MKI$(IN.DEL.PTR)
```

```
332 LSET IN.DEL.COUNT$=MKI$(IN.DEL.COUNT)
333 PUT 5,1:CLOSE 5
340 FIELD 4,2 AS ALLOC.REC$,2 AS LAST.REC$,
    2 AS LRL$,LRL-6 AS DUMMY$
350 GET 4,ALLOC.REC+1
360 LSET LAST.REC$=MKI$(LAST.REC)
370 PUT 4,ALLOC.REC+1
380 CLOSE
390 END
1000 REM Reuse Deleted Records
1010 GET 4,DEL.PTR 'read in the deleted record
1020 PREV.DEL$=OUT.BUFF$(3)+OUT.BUFF$(4)
     'get location of next deleted record
1030 PREV.DEL=CVI(PREV.DEL$)
1035 DEL.COUNT=DEL.COUNT-1
1040 IF DEL.COUNT=0 THEN PREV.DEL=0
     'if no more deleted records set to zero
1050 GOSUB 3000
1060 PUT 4,DEL.PTR
1070 DEL.PTR=PREV.DEL
1080 RETURN
2000 REM Use Additional Records
2010 LAST.REC=LAST.REC+1
     'pointer to next record to use
2020 IF LAST.REC > ALLOC.REC THEN ERROR 63
     ELSE GOSUB 3000
2030 PUT 4,LAST.REC
2050 RETURN
3000 REM Move input record to output record
3010 FOR SETUP=1 TO LRL
3020 LSET OUT.BUFF$(SETUP)=IN.BUFF$(SETUP)
3030 NEXT SETUP
3040 RETURN
5000 REM Now delete the transferred record
5010 LSET IN.BUFF$(1)=CHR$(ASC(IN.BUFF$(1))+128)
5015 IF IN.DEL.PTR=0 THEN IN.DEL.PTR=-1
5020 LSET IN.BUFF$(3)=LEFT$(MKI$(IN.DEL.PTR),1)
5030 LSET IN.BUFF$(4)=RIGHT$(MKI$(IN.DEL.PTR),1)
5040 PUT 3,IN.REC+1
5050 IN.DEL.PTR=IN.REC+1
5060 IN.DEL.COUNT=IN.DEL.COUNT+1
5070 RETURN
6000 IF ERR=63 AND ERL=2020
     THEN LAST.REC=LAST.REC-1:PRINT "There are not
     enough records allocated in "
     OUT.FILE$".":RESUME 300
6001 PRINT "Unexpected error "ERR" in line "ERL
6002 END
```

Table 1. LB Data Pointers and Counters

All of the variables mentioned in this table exist as both integer variables and random-access-File field variables. The field variables are differentiated by the suffix $.

Variable	Purpose and File Location
ALLOC.REC	Contains the number of pre-allocated records in a Little Brother Database; may be found in sector 1, bytes 27-28 of /DEF file and in the last sector, bytes 1-2 of the /LB file.
LAST.REC	Contains the highest numbered record in the /LB file that has been used to store a database record; may be found in sector 1, bytes 29-30 of /DEF file and in the last sector, bytes 3-4 of the /LB file.
LRL	Contains the number of bytes in each /LB data record; may be found in sector 1, bytes 31-32 of /DEF file and in the last sector, bytes 5-6 of the /LB file.
DEL.PTR	Contains the number of the last record that was deleted, or zero, if no records have been deleted; found in sector 1, bytes 33-34 of /LB file.
DEL.COUNT	Contains the count of delete records; found in sector 1, bytes 90-91 of /LB file.
IN.REC	Contains the number of records that were selected for the active index; stored in bytes 1-2 of the /SLx file.



The Programmers' Corner

The following programs, although shown in their respective source code language, are nevertheless applications which may be directly usable by the non-programming user. All you need do is obtain the assembled/compiled program from the DISK NOTES 3.1 diskette which is associated with this issue of THE MISOSYS QUARTERLY.

Popup Application Window by Roy Soltoff

This utility, which I have named "PAW", provides a single keystroke execution of any PRO-WAM application directly from a PRO-WAM library file. Although the utility of the program is strictly for the PRO-WAM user, I have developed versions of the program in three different computer languages. The actual implementation of the program versions should be informative in and of themselves; therefore, I will spend most of this article discussing the implementation; the use is rather straightforward.

Versions of PAW are programmed in ASSEMBLER using the PRO-CREATE or PRO-MRAS assemblers; C using the PRO-MC compiler; and BASIC using the PRO-EnhComp compiler. Each implementation follows more or less the same algorithm. Because of this, it would be proper to present the algorithm first. However, I need to at least note a few peculiarities of the PRO-WAM environment.

A PRO-WAM library file, say WAM0/APL, has a directory which is used by the resident PRO-WAM application loader module to locate the name and position of each application contained in the library. When I implemented the directory scheme, its design had to be extremely simplified. It was important to minimize the high memory used by the loader; thus, the shortest possible code should be written to load a member. That necessitates a simplified scheme.

An application has up to eight characters for its file name. I limited the library directory to one 256-byte sector; thus, a library can contain at most 32 applications (members); a member name appears every eight character positions. Therefore, a single READ can obtain the directory.

I further simplified the access by using a single member size fixed for all applications. Each application is at most nine sectors long (that's 8 for the code and 1 for the header). This means that the library file position of a member is known directly by the location of its name in the directory. The directory is sector 0; members are at sectors 1, 10, 19, 28, etc. This makes it easy to calculate the location of the header by [DIRLOC + (DIRLOC/8) + 1] where DIRLOC is the offset of the member name in the directory (0, 8, 16, 24, etc.)

I make use of the @WEXEC function which was added in PRO-WAM release 2. This function allows you to invoke an application under programmed control. The only information passed to the function is a pointer to the member specification string [mspec$]. The format of mspec$ is discussed in the PRO-WAM user manual. It consists of an optional library number, the member name, and an optional drive specification. The library number will default to zero if omitted. But the drive specification will default to the drive last used to access a PRO-WAM application. Therefore, the program builds a complete member specification string. It does this by using the library number you designate to open the library file, then it examines the file control block to obtain the number of the drive where the DOS opened the file. It then uses this information to build the mspec$ to pass to @WEXEC.

PAW is invoked via the following command line:

```
PAW lib#
```

where "lib#" is an optional single character designation for a library. If omitted, it will default to "0". A complete procedure for each programmed version of PAW is presented in the 12 steps which follow:

1. Examine the command line for an argument. If none entered, default the library requested to "0". If entered, check for validity as PRO-WAM library files can be numbered from "0" through "9".

2. Open the designated library file. Abort if any error is detected on the open attempt.

3. Save the number of the drive where the file was opened.

4. Read the directory sector.

5. Close the library file.

6. Clear the screen and issue the welcome message.

7. Search through the directory and display the names of all members. Assign each member a one-character designator to be used to invoke that member. Display names on the screen four to a row.

8. Issue the selection message.

9. Accept a single character which is to designate a member. BREAK is used to exit PAW.

10. Convert the character input to upper case to ease the procedure of matching against the character assigned in step 7.

11. If a match is found, build the member specification string and @WEXEC the requested application.

12. Go to step 9.

In order to more easily discuss each program implementation, I have provided a 3-digit line number for each line in the program. The line numbering was easily added by using the number C program which follows. Thus a simple,

```
NUMBER <PROGIN >PROGOUT
```

will prefix each line with a 3-digit number starting from 1 followed by a colon then a space.

```
/* number.c */
#include <stdio.h>
char linebuf[256]; int line=0;
main()
{  while (fgets(linebuf,255,stdin))
        printf("%03d: %s",
++line,linebuf);
}
```

I choose to discuss the assembler version of PAW first only because it is the most straightforward. There is nothing fancy in it. In implementing the C version, I have used a structure which documents the file control area (FCA) of MC streams. This has never been documented before. Most of the data obtainable from the structure can be obtained from fstat(). On the other hand, you may want direct access to the data in the FCA for your own reasons. In any event, that's what's special about PAW/CCC. In implementing the EnhComp BASIC version, I have programmed a number of statements which merit observation. I make heavy use of the REPEAT-UNTIL construct; I have used user-defined COMMANDS and multi-line FUNCTIONS; I have used in-line assembly code; and I have provided a technique whereby you can parse command line parameters from the compiled program.

As a matter of interest, I coded the C version first. I then coded the assembler version. Finally, I coded the EnhComp BASIC version. The easiest for me was probably the assembler version. The fastest to code was probably the C version; however, the assembler version was pretty close. I just happen to be a little more expert in assembler than C. Since I rarely code in BASIC, that version took the longest. I also had to program more routines to extract information which was unavailable. It was also the most frustrating because I came across a few bugs in the EnhComp compiler which caused it to bomb when handling some error reporting (if I had no program errors, I wouldn't have had the bomb!). Those bugs and the respective fixes are printed in this issue's *MISOSYS Products' Tidbits*. I also had a problem with a PRO-WAM patch I was working up. Everything seemed to go wrong. On the other hand, I may have had the most fun with the BASIC version since I had to learn the most while coding it.

In the following listings, I have added substantial comments to explain the approach taken. These comments are not part of the source files. Therefore, they are not line-numbered. In order to avoid confusing the added comments from program lines which may wrap around, I will prefix each comment with a period when appearing within a block of code. Major comments will be added external to a block of code.

PAW/ASM - Assembler version

This first section defines various constants used in the program as well as SVC assignments. The approach of using a macro to expand SVC calls is common in assembler listings; however, I normally use more elaborate macros in my personal programs. The simplified macro here just simplifies the program listing a little and gets you into the habit of considering the use of macros in your programs.

```
001: ;PAW/ASM - Popup Application Window - 07/20/88
002: ETX          EQU       3
003: CR  EQU      13
004: LF  EQU      10
005: BREAK        EQU       80H
006: @DSP         EQU       2
007: @KEY         EQU       1
008: @DSPLY       EQU       10
009: @VDCTL       EQU       15
010: @ERROR       EQU       26
011: @OPEN        EQU       59
012: @CLOSE       EQU       60
013: @READ        EQU       67
014: @CLS         EQU       105
015: @WINDOW      EQU       124
```

This is the simplified macro prototype. It just expands the SVC number into the two operations between the MACRO and ENDM pseudoOPs.

```
016: SVC          MACRO     #OPCODE
017:      LD      A,#OPCODE
018:      RST     28H
```

```
019:      ENDM
```

The program origin and the resulting size of this assembler version program allow it to execute solely within the library memory region of the DOS. Thus, it's suitable for invocation from LIBEXEC.

```
020:      ORG       2600H
```

When a program begins execution, the DOS sets up certain program entry conditions. Register HL points to the first non-blank character following the program name. This portion first executes. It checks the command line character. If a library number was entered, then the character would be something other than a RETURN (13D). Otherwise, the character is inserted into the application library file specification and then checked for validity. Note that a single comparison is used for this test. Since the value under test in the accumulator need not be preserved, it takes up less code space to test a single range of values by adjusting the base to zero. Thus, the SUB '0' instruction alters the valid range of character values from '0'-'9' to 0-9. Note the former are ASCII values 30H-39H while the latter are 00H-09H.

```
021: BEGIN        LD        A,(HL)      ;Get command
line char
022:      CP        CR
023:      JR        Z,NOLIB
024:      LD        (FAPP$+3),A         ;Set possible
libnum
025:      SUB       '0'                 ;  into fspec
and
026:      CP        9+1                 ;  check
validity
027:      JP        NC,BADLIB
```

The fourth character of the file specification contains the library number either established by default or altered by a comman line parameter. This value is saved for later use in building up the member specification.

```
028: NOLIB        LD        A,(FAPP$+3)         ;Save
LIBnum for later
029:      LD        (LIB),A
```

The library file is opened. A branch is taken if an error is returned from the OPEN function.

```
030:      LD        DE,FAPP$            ;Open the
library file
031:      LD        HL,BUFFER
032:      LD        B,0
033:      SVC       @OPEN
034:      JP        NZ,NOFIND           ;Go on error
```

The seventh byte of the file control block contains the number of the drive (in binary) containing the opened file. This value is recovered and converted to ASCII whereby it is saved for later use in building up the member specification.

```
035:      LD        A,(FAPP$+6)         ;Get drive
number
036:      OR        '0'                 ;  where found
and
037:      LD        (DRIVE$),A          ;  save in ASCII
```

The first sector which contains the directory is read. If an error occurs, an appropriate branch is taken. Otherwise, the file is then closed and the welcome message is displayed.

```
038:      SVC       @READ               ;Read the dir
record
039:      JP        NZ,IOERR
040:      SVC       @CLOSE              ;Close the file
041:      CALL      WELCOME             ;Screen heading
```

I am using a 32-byte array [CODE$] to store the single character code letter assigned to each application member. If a directory position has no member assigned to it, WAMLIB will fill that position with spaces. That's what I use to test for an APP. The screen display sets up a maximum of four names per row starting at the ninth column. This means I have to tab over to column 8 (based 0) every fourth name displayed. The tabbing routine also emits a RETURN to end the previous line.

```
042:      LD        IX,CODE$            ;Point to code
table
043:      LD        HL,BUFFER           ;Point to buffer
start
044:      LD        B,0                 ;Init counter
for tabbing
045: SCAND        LD        A,(HL)      ;Get 1st char
of APPNAME
046:      CP        ' '                 ;If blank, no
APP there
047:      JR        Z,SCAND2
048:      LD        A,B                 ;Do we tab here?
049:      AND       3                   ;Tab at start of
050:      CALL      Z,TAB8              ;  every fourth
APP
051:      LD        C,'['               ;Display '[code]
'
052:      SVC       @DSP
053:      LD        A,(LETTER)
054:      LD        (IX),A              ;Save code for
this one
055:      LD        C,A
056:      SVC       @DSP
```

```
057:      LD        C,']'
058:      SVC       @DSP
059:      LD        C,' '
060:      SVC       @DSP
061:      INC       B                    ;Bump counter
062:      PUSH      HL
063:      PUSH      BC                   ;Preserve
counter
064:      LD        B,8
065: SCAND1       LD        C,(HL)     ;Display
APPNAME
066:      INC       HL
067:      SVC       @DSP
068:      DJNZ      SCAND1
069:      CALL      TAB4                 ;Now four spaces
070:      POP       BC
071:      POP       HL
```

At this point, I have completed the processing of the current directory "slot". Now I increment the pointer to my CODE$ array for the next position. I then increment the letter value. Since there can be up to 32 entries, I must test for when the letter value gets past "Z"; thus, the special test switches the codes to the numbers "0" through "5" for the six cases of slots 27-32. Note that in this assembler implementation, I have bypassed code letters when a directory slot is vacant. As you will discover by examining the other two implementations, I don't skip letters because the code letter incrementing is done only when I have found a member. Thus, the implementations aren't exact; however, it demonstrates different ways of doing something.

```
072: SCAND2       INC       IX          ;Bump code
pointer
073:      LD        A,(LETTER)           ;Bump code
letter
074:      INC       A
075:      CP        'Z'+1                ;Switch to 0-9
076:      JR        C,SCAND3             ;  after A-Z
077:      LD        A,'0'
078: SCAND3       LD        (LETTER),A
```

After processing for a directory entry, I advance to the next one. When you are working with a 256-byte buffer, it is extremely useful to origin the buffer at a page boundary (that's a 16-bit memory address where the low-order 8 bits are zero). Since 8-bit ADD instructions set the CARRY flag on overflow, a simple ADD of 8 to the low-order register containing BUFFER not only advances to the next directory position, but also lets me know when I have reached the end of the buffer. I loop through the thirty two entries then emit the selection query.

```
079:      LD        A,L                  ;Bump to next
080:      ADD       A,8                  ;  slot
081:      LD        L,A
082:      JR        NZ,SCAND             ;Loop if more
083:      CALL      SELECT
```

A keystroke is accepted, converted to upper case since I have used upper case letters for the codes, then tested for BREAK - which is used to designate an EXIT of the program. I then search through the CODE$ array for a match to the character entered. If no match is found, I go back and accept another character. Since the entry is not displayed, the only means of noting your selection is when the designated application executes.

```
084: ;*=*=*
085: ;   Select application code letter
086: ;*=*=*
087: INPUT        SVC       @KEY       ;Get a
keystroke
088:      CALL      TOUPPER              ;Convert to
upper case
089:      LD        B,32                 ;Scan code list
090:      LD        HL,CODE$
091:      CP        BREAK                ;Break key?
092:      JR        Z,EXIT
093: INPUT1       CP        (HL)       ;A match?
094:      JR        Z,INPUT2
095:      INC       HL                   ;Bump to next
code and
096:      DJNZ      INPUT1               ;  check further
097:      JR        INPUT                ;Oops, no more
codes
```

Based on the value remaining in the loop counter, I can calculate the position of the member name in BUFFER.

```
098: INPUT2       LD        A,32       ;Develop index
into
099:      SUB       B                    ;  the APPNAMEs
100:      ADD       A,A                  ;x 2
101:      ADD       A,A                  ;x 4
102:      ADD       A,A                  ;x 8
103:      LD        H,.HIGH.BUFFER       ;Set up APPNAME
pointer
104:      LD        L,A
```

Here's where I build the member specification, mspec$. The library number has already been placed into the first position. I now insert characters from the directory BUFFER until either the first space is reached or eight characters have been transferred. I then add the drive specification and terminate the string with a NULL. Actually, PRO-WAM stops on any character value less than a SPACE after fetching the string with the @FSPEC SVC. Of course, @FSPEC will terminate its scanning on any character not alphanumeric unless its a valid file specification character so you can use RETURN, ETX, or NULL.

```
105:      LD        DE,MSPEC$+1          ;Bypass LIBnum
106:      LD        B,8                  ;Init for 8
chars max
107: INPUT3       LD        A,(HL)     ;Transfer until
either
```

```
108:      LD        (DE),A              ;  eight
characters or
109:      CP        ' '                 ;  until a SPACE
has
110:      JR        Z,INPUT4            ;  been sensed
111:      INC       HL
112:      INC       DE
113:      DJNZ      INPUT3
114: INPUT4         LD        A,':'      ;Now insert
drive
115:      LD        (DE),A
116:      INC       DE
117:      LD        A,(DRIVE$)
118:      LD        (DE),A
119:      INC       DE
120:      XOR       A                   ;Terminate mspec
121:      LD        (DE),A
```

Now invoke the requested application via the @WEXEC function of PRO-WAM's WINDOW SVC. Then loop back to fetch another request until BREAK is detected. Note that since the screen is preserved by PRO-WAM, we don't have to re-draw it.

```
122:      LD        HL,MSPEC$
123:      LD        B,12
124:      SVC       @WINDOW
125:      JR        INPUT               ;Loop until
BREAK
```

This exit branch is taken when the input routine detects the BREAK key. Here I just clear the screen and return to DOS. Note the return code is set to 0 in this case which indicates no error.

```
126: EXIT           SVC       @CLS
127:      LD        HL,0
128:      RET
```

This routine posts an error message when an invalid library number is entered. It then returns to DOS. Note the return code is set to -1 in this case which indicates an error.

```
129: BADLIB         LD        A,(FAPP$+3)
130:      LD        (BADLIB$+9),A
131:      LD        HL,BADLIB$
132:      SVC       @DSPLY
133: ERREXIT        LD        HL,-1       ;Indicate error
exit
134:      RET
```

This routine processes any error code returned from DOS when opening the application library. I specifically test for the "File not found" error to display my own error message. Why? No particular reason, but it shows you how to do it.

```
135: NOFIND         CP        24          ;File not
found?
136:      JR        NZ,IOERR
137:      LD        HL,NOFIND$
138:      SVC       @DSPLY
139:      JR        ERREXIT
```

This is a standard error handling routine which uses the DOS error service call to post the appropriate message. Most folks tend to force the abbreviated error message which just displays short strings. Sometimes, the message is insufficient to get at the root of the problem because the exact file names or drives (where appropriate) aren't displayed. Ever get a "Parity error during read" during a BACKUP. Think it was on the SOURCE disk? Think again! During a BACKUP operation, reading is done on both the SOURCE and DESTINATION. On a backup by class or reconstruct, you will be hard pressed to note which drive was being read when that message gets displayed. You can always override any forced abbreviation by setting bit 6 of the system flag. That can be done from DOS Ready simply by invoking the command,

```
MEMORY (A="S",B=64)
```

Of course, any other bits will be reset by this so you may want to check the value first and then add 64 to it.

```
140: IOERR          OR        0C0H        ;Mask abbrev,
no abort
141:      LD        C,A
142:      SVC       @ERROR              ;Post error
143:      JR        ERREXIT
```

The welcome routine clears the screen, then positions the cursor to column 15 because that's where I calculated it to start if the message is to be centered. I could have eliminated the cursor positioning and added 15 spaces to the message; but this way is shorter in memory.

```
144: ;*=*=*
145: ;   Routine to issue welcome heading
146: ;*=*=*
147: WELCOME        SVC       @CLS        ;Clear the
screen
148:      LD        HL,15               ;Set cursor to
column 15
149:      LD        B,3
150:      SVC       @VDCTL
151:      LD        HL,HELLO$
152:      SVC       @DSPLY
153:      RET
```

Nothing special here. Just a routine which returns twice (once in the TAB8 routine) then issues a message spaced over to column 8.

```
154: ;*=*=*
155: ;   Routine to issue selection prompt
156: ;*=*=*
157: SELECT      LD       C,CR       ;1st a CR
158:     SVC     @DSP
159:     CALL    TAB8
160:     LD      HL,SELECT$
161:     SVC     @DSPLY
162:     RET
```

These routines merit some attention. I wanted a routine to TAB over 4 spaces and one 8 spaces. Here is a classic example of one routine calling the trailing portion of itself to repeat part of it then falling into the code just called. You'll find such a routine useful for all sorts of things. The tradeoff is slightly longer execution time caused by the CALL-RET handling against reduced code space. When memory is a premium, as it is with a 64K address space, it is generally better to optimize for space than for execution unless you are writing an interrupt task where fast timing is critical.

```
163: ;*=*=*
164: ;   Routines to tab 8 or 4 spaces
165: ;*=*=*
166: TAB8        LD       C,CR       ;1st a CR
167:     SVC     @DSP
168:     CALL    TAB4               ;Tab 4 spaces
then 4 spaces
169: TAB4        PUSH     BC         ;Tab 4 spaces
170:     LD      B,4
171: TAB4A       LD       C,' '
172:     SVC     @DSP
173:     DJNZ    TAB4A
174:     POP     BC
175:     RET
```

Here's a classic routine to convert characters in the range "a-z" to their upper case counterparts "A-Z". Any character value outside the lower case range is untouched.

```
176: ;*=*=*
177: ;   Routine to convert char in A to upper case
if lower
178: ;*=*=*
179: TOUPPER     CP       'a'
180:     RET     C
181:     CP      'z'+1
182:     RET     NC
183:     SUB     20H
184:     RET
```

I usually put my data last. Some folks put it first. If you write code for MS-DOS, you'll make every effort to put your data first. Note that I have EQUated "LIB" to MSPEC$. This allows me to use a discrete name for the library number yet store it as the first character of the member specification string. I could have referred to it solely by MSPEC$, but this way it allows the three different implementations of PAW to use similar variable names where possible.

```
185: MSPEC$      DB       '0WAPPNAME:d',CR
186: LIB         EQU      MSPEC$
187: DRIVE$      DB       0          ;APL drive
number
188: LETTER      DB       'A'        ;Code for APP
189: CODE$       DC       32,0
190: HELLO$      DB       'Popup application window
for PRO-WAM applications',CR
191: SELECT$     DB       'Select your application;
<BREAK> to exit...  ',ETX
192: BADLIB$     DB       'Library ''x'' is
invalid...',CR
193: NOFIND$     DB       'Cannot locate
application library: '
194: FAPP$       DB       'WAM0/APL',CR
195:     DC      32-$+FAPP$,0       ;Expand FCB to
32
```

This is where I define the BUFFER to be on a page boundary. The conditional test handles the one time out of 256 that the program counter would be already at a page boundary. In that case, I don't want to skip a page.

```
196:     IF      $.AND.255
197:     ORG     $.AND.0FF00H+256
198:     ENDIF
199: BUFFER      DS       256
200:     END     BEGIN
```

PAW/CCC - C Language version

I start off the C language by including the header files associated with MC library functions used in the program. The INLIB option is designated to request a search of the IN/REL library; that's where I installed the WINDOW support routines.

```
001: /* paw - Popup Application Window for PRO-WAM
*/
002: #include <stdio.h>
003: #include <fcntl.h>
004: #include <sgtty.h>
005: #include <memory.h>
006: #option INLIB     /* need for window/rel */
```

Here's something new for you MCers. It is a data structure of the file control area used in MC streams. As an aside, the FCA and its association with FILE pointers was first used in LC and developed by Jim Frimmel - the designer of LC. At that

time, the FCA consisted of a single byte containing eight flags, a 32-byte region for the DOS File Control Block (FCB), and a one byte buffer to store a single push-back character or last error code encountered.

LC maintained a table of pointers which was called the File Vector Table, FVTBL. When I revised the MC stream functions to better support UNIX System V standards, it required more space in the FCA. The ioctl() function documents the control word added and is generally used to access the first three bytes of the expanded FCA block. Since all of the stream functions were written in assembler, and the indexing of the FCA was based on the file pointer, it was less error prone from a programming standpoint to just tack on two bytes to the front of the FCA and not alter the file pointer value. When you fopen() a file, an FCA block is allocated and optionally a file buffer. The address of the control flag is inserted into a vacant FVTAB slot. Note that this address is two bytes beyond the first byte allocated. The address of the FVTAB slot is returned as the FILE pointer. Ideally, I should have altered the coding in all of the stream functions to keep the FILE pointer address the same as the structure address; however, since it was all internal, it really didn't matter. I was more concerned with introducing coding errors (bugs!).

As another aside, MC uses the FVTAB slots as FILE DESCRIPTORS. Recollect that DESCRIPTORS are small numbers. By default, STDIN is usually 0, STDOUT is 1, STDERR is 2, etc. Thus, the sequential slots in the File Vector Table represent FILE DESCRIPTORS 0-n. A conversion function such as fileno() simply subtracts the table origin from your file pointer and divides the result by two to arrive at the associated file descriptor.

```
007: /* MC file control area (FILE *) */
008: struct fca
009: {   unsigned short    control;   /* 16-bit
control field */
010:             char      flag;      /* fca flag */
011:             char      type;      /* FCB type */
012:             char      iostat;    /* input/output
status */
013:             char      pdsmoo;    /* PDS member
origin offset */
014:             char      *iobuf;    /* pointer to
i/o buffer */
015:             char      nrnoff;    /* NRN offset
*/
016:             char      drive;     /* logical
drive number */
017:             char      dec;             /*
directory entry code */
018:             char      ernoff;    /* ERN offset
*/
019:             char      lrl;             /*
logical record length */
020:     unsigned int  nrn;           /* next record
number */
021:     unsigned int  ern;           /* ending
record number */
022:     unsigned int  sext;        /* starting
extent */
023:             long      equad[4];          /*
extent quads */
024:             char      cbec;      /* 1-char buf
or error code */
025: };
```

Here's some data declarations. Remember, when you don't need automatic variables, its best to make them global or static. Execution is faster when MC doesn't have to access variables on the stack. I'm using a structure for the directory buffer to more easily recognize the indexing. I could have used a two dimensional array, but structures are more clear. The code array is declared and each element initialized to zero. That's sort of like the "CODE$ DC 32,0" in the assembler version, but since MC doesn't know about DC, it generates 32 DB statements. Of course, the end result in the executable file is the same.

```
026: char *fapp = "wam0/apl";
027: char letter='A', lib='0', mspec[12], *pmspec;
028: FILE *stream;
029: struct sgttyb sg;
030: struct {
031:          char name[8];
032:      } buffer[32];
033: char code[32] =
034:
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0},
035:  appname[9] = {0,0,0,0,0,0,0,0,0};
036: struct fca *fca;
```

Main is declared with the typical argc and argv arguments to be able to gain access to any command line parameter. I then declare the remaining variables needed by the program to be statics - again for speed of access.

```
037: main(argc,argv) int argc; char *argv[];
038: {   static int i, appfile, appmax, currow,
curcol, drive, count;
```

If there is a parameter on the command line, it is assumed to be the library number. Here I get that character, save it, then test it for validity. If valid, I insert it into the application library file specification, otherwise, I abort with an appropriate error message.

```
039:     if (argc > 1)
040:     {
041:          lib = *argv[1];
042:          if (lib < '0' || lib > '9')
043:          {
044:               printf("Library '%c' is
invalid...\n",lib);
045:               exit(-1);
046:          }
```

```
047:          *(fapp+3) = lib;
048:     }
```

The library file is opened using the block I/O function and tested for an error in opening. If an error is detected, the program will abort with an appropriate error message

```
049:     if ((appfile=open(fapp,O_RDONLY)) == EOF)
050:     {
051:          printf("Cannot locate application
library: %s\n",fapp);
052:          exit(-1);
053:     }
```

In order to obtain the drive number directly from the FCA structure, I have to first get the file pointer. The fpup() function is used for this.

```
054:     /* get file pointer */
055:     if (!(stream = fpup(appfile)))
056:          return EOF;
057:
```

Line 59 is Rich Deglin's code to fetch the address of the fca structure. Note the use of casts. I then capture the drive number, convert it from binary to ASCII, and store it.

```
058:     /* Get struct fca pointer */
059:     fca = (struct fca *)((char *) *stream -
sizeof(unsigned short));
060:     drive = fca->drive + '0';
```

The directory sector is read into the buffer. Appropriate action is taken if an error is detected, other wise the file is closed.

```
061:     errno = 0;
062:     if ((read(appfile,buffer,256))!=256)        /
* get the app directory */
063:          {
064:               perror("Application library read
error");
065:               exit(-1);
066:          }
067:     close(appfile);
```

Under MC, BREAK can be set to generate an EOF condition on a stream. This next sequence of statements conditions this behavior via the ioctl() function. Then the welcome message is displayed.

```
068:     /* Turn on BREAK to set EOF */
069:     ioctl(STDIN,TIOCGETP,&sg);
070:     sg.sg_control |= IO_BREAK;
071:     ioctl(STDIN,TIOCSETP,&sg);
072:
073:     welcome();                    /* Headline
message */
```

This loop examines the 32 directory slots. C's modulo operator is used to help generate the RETURN TAB8 operations at every four member names displayed (as well as prior to the first one). In this implementation, code letters will be skipped for vacant directory slots as the loop index is used to generate the code letter rather than the count of members displayed. This also readjusts the range of codes to be 'A-Z', '1-6'. After the directory has been scanned, the selection query is posted.

```
074:     for (i = count = 0; i<32; i++)
075:     {
076:          if (buffer[i].name[0] != ' ')
077:               {
078:               if (!(count % 4))
079:                    fputs("\n\t",stdout);  /*
start at column 8 */
080:               printf("[%c] %-8s
",letter+i,memcpy(appname,buffer[i].name,8));
081:               code[i] = letter+i;
082:               ++count;
083:               }
084:          if (i == 25)
085:               letter = '1'-26;
086:     }
087:     printf("\n\n           Select your
application; <BREAK> to exit...  ");
```

The WHILE loop is forever since its conditional test is a constant TRUE. The detection of EOF on the key request is the means of escape. A simple for loop compares the requested code letter against the list previously built. I have used the memccpy() routine to perform the transfer of the member name into the member specification. That function will stop after the test character has been transferred or until the limiting number of characters have been transferred. The building up of the spec is straightforward. The application is invoked via the wexec() function and the "break" statement escapes the for loop only to pass control to the beginning of the while loop.

```
088:     while (1)
089:     {    if ((letter = getchar())== EOF)
090:               break;
091:          letter = toupper(letter);
092:          for (i = 0; i <= 32; i++)
093:          {
094:               if (code[i] == letter)
095:               {    pmspec = mspec;
096:                    *pmspec++ = lib;
```

```
097:                    if (!( pmspec = memccpy(
pmspec, buffer[i].name,' ',8)))
098:                        pmspec=mspec+9;
099:                    *--pmspec=':';
100:                    *++pmspec=drive;
101:                    *++pmspec=0;
102:                    wexec(mspec);
103:                    break;
104:                }
105:            }
106:        }
```

A BREAK key depression generates the EOF which escapes from the while loop. Here the screen is cleared and the program ends. The remaining functions support main() and do not merit further attention.

```
107:        cls();
108: }
109: welcome()
110: {   cls();
111:        cursor(15,0);
112:        puts("Popup Application Window for PRO-WAM
Applications");
113: }
114: cls()
115: {   puts("\x1c\x1f");      }
```

PAW/BAS - EnhComp BASIC version

This BASIC version of PAW was written for the PRO-EnhComp compiler. I at first questioned myself about writing a BASIC version, then thought that it would shed a little more light on the use of EnhComp. Besides, I started wondering how I would capture some of the data referenced in the other two versions. That gave me the impetus to work in BASIC. Of course, I wouldn't even begin to try interpretive BASIC, although I would suspect that it could be used as well except for getting command line parameters.

BASIC allows a common facility for error handling. Here's where I first establish the routine to take control if an error is detected. Note the clarity of code evidenced by the use of line labels in lieu of BASIC line numbers. The ALLOCATE statement is used to establish a file buffer; EnhComp defaults to none. The rest of this block is straight BASIC.

```
001: ON ERROR GOTO "IOERROR"
002: DEFINT A-Z:CLEAR 1000
003: ALLOCATE 1:CLS:
DIM CODE$(32):LETTER = ASC("A")
004: FAPP$="wam0/apl":LIB$="0"
```

Line 5 assigns the result of the command line argument fetch to the variable T$. The GETARG$ function is a user defined multiline function defined later. We'll discuss it then. For now, all I need say is that it returns a sting value. The "!" character tells the EnhComp compiler that a multiline function call follows. The rest of the block is straightforward BASIC; however, I have made use of the block-structured IF/ENDIF. I think this exhibits far greater clarity than the confusing IF/THEN/ELSE of interpretive BASIC.

```
005: T$ = !GETARG$
006: IF T$ <> ""
007:     IF T$ < "0" OR T$ > "9"
008:         PRINT "Library ";
CHR$(27);T$;CHR$(27);
" is invalid..."
009:         END
010:     ENDIF
011:     MID$(FAPP$,4,1) = T$:LIB$=T$
012: ENDIF
```

Here, the application library file is opened and fielded. Note that I can field the complete 256-byte sector into a single string of 256-byte length. EnhComp supports strings up to 32767 characters long which is more than sufficient for a 64K machine.

You also see another user-defined multiline function here. GETDRV$ is used to obtain, as a one-byte ASCII string, the number of the drive which contains the library file just opened. More on GETDRV$ later. Note that it does have an argument, the file buffer number.

After the directory sector is read, the file is closed and the welcome message is posted.

```
013: 100 OPEN "R",1,FAPP$
014: FIELD 1, 256 AS BUFFER$
015: 110 GET 1,1
016: DRIVE$=!GETDRV$(1)
017: CLOSE 1
018: PRINT TAB(15);"Popup Application Window for
PRO-WAM Applications"
```

This FOR-NEXT loop scans the directory buffer, builds up the array of code letters, and displays the names of the library members. In this example, code letters are always assigned sequentially with no gaps because the variable used to count the number of displayed member names is used to generate the code letter. It's just a different way of building code letters.

This block also uses the structured IF/ENDIF. It also uses the integer incrementing command, INC, which would be equivalent to the statement "COUNT = COUNT + 1".

```
019: COUNT = 0
020: FOR I = 0 TO 31
```

```
021:     IF MID$(BUFFER$,8*I+1,1) <> " "
022:         IF (COUNT AND 3) = 0
023:             PRINT:PRINT TAB(8);
024:         ENDIF
025:         PRINT "[";
CHR$(LETTER+COUNT);"] ";
MID$(BUFFER$,8*I+1,8);"    ";
026:         CODE$(I) = CHR$(LETTER+COUNT)
027:         INC COUNT
028:         IF COUNT = 25 THEN LETTER = ASC("1")-26
029:     ENDIF
030: NEXT I
031: PRINT:PRINT:PRINT TAB(8);
"Select your application; <BREAK> to exit...  ";
```

With the directory sector scanned and displayed, I need to establish a procedure for handling the exit escape. Since I have used a BREAK in the other two versions for this operation, I will use that here as well. This is a fine use for the "ON BREAK GOTO" facility of EnhComp. How would you do that with interpretive BASIC?

```
032: ON BREAK GOTO "EXIT"
```

This loop is also a forever loop with the construct "REPEAT expression UNTIL 1 <> 1"; which rarely occurs! The WINKEY$ function in EnhComp is similar to the repeated INKEY$ until a non-null string results. I used another multi-line user defined function here to convert the entered code letter request to upper case. The inner REPEAT-UNTIL loop is used to calculate the length of the member name. I could have used a FOR-NEXT loop, but they are kind of messy in compilers when you want to exit the loop prematurely. Note how I set the loop index, I, to a value of 31 to cause a premature exit of the FOR-NEXT loop in this block after the WEXEC command is executed. That's the correct way to prematurely terminate FOR-NEXT loops.

The "%" prefix in front of WEXEC tells the compiler that a user command follows. And you didn't even know that BASIC was extensible!

```
033: REPEAT
034:     LETTER$=WINKEY$
035:     LETTER$ = !TOUPPER$(LETTER$)
036:     FOR I = 0 TO 31
037:         IF CODE$(I) = LETTER$
038:             MSPEC$ = LIB$
039:             T$ = MID$(BUFFER$, 8*I+1, 8):MLEN=9
040:             REPEAT
041:                 DEC MLEN
042:             UNTIL MID$(T$,MLEN,1) <> " "
043:             MSPEC$ = LIB$ + LEFT$(T$,MLEN) +
":" + DRIVE$ + CHR$(13)
044:             %WEXEC(MSPEC$)
045:             I=31
046:         ENDIF
047:     NEXT I
048: UNTIL 1 <> 1
```

This is the routine entered on detection of the BREAK. It's a simple labeled block which clears the screen and returns to DOS.

```
049: "EXIT"
050: CLS:END
```

Here's the error handler. Since line numbers are not needed for EnhComp statements, I have added line numbers only to the lines which would expect an error which I wish to report on. I then calculate a GOSUB index based on the line number returned from ERL.

```
051: "IOERROR"
052: ON (ERL-90)/10 GOSUB "ERR1", "ERR2"
053: END
054: "ERR1"
055: PRINT "Cannot locate application library:
";FAPP$
056: RETURN
057: "ERR2"
058: PRINT "Application library read error"
059: RETURN
```

Here's the function which returns the number of the drive associated with the file buffer passed as the argument. The drive number is returned as an ASCII string. Astute readers will reconize a portion of this function as being identical to the SETEOF() function covered in an early issue of THE MISOSYS QUARTERLY as well as included with the 2.6 release of EnhComp.

The function first establishes a one-byte string for the return value since its far easier to let HIGH-MODE allocate the string space. It then switches to Z80-MODE and uses assembly language; it calls the library support routine which obtains the address of the file buffer control area after first testing the validity of the buffer number. The address of this area is returned in register IX. The control structure is documented on page 5-2 of the EnhComp manual.

I then pick up the drive number and convert it to an ASCII character. Next, I point IX to the address of the string control block for "T$". Register HL is then loaded with the address of the string and the contents of that one location are loaded with the ASCII drive number. I then switch back to HIGH-MODE and return the value of T$ as the function value.

```
060: FUNCTION GETDRV$(BUFNUM%)
061: T$="0"
062: Z80-MODE
```

```
063: LD HL,(&(BUFNUM%)):CALL @CALADR
064: LD A,(IX+16+6):ADD A,'0'
065: LD IX,&(T$)
066: LD H,(IX+3):LD L,(IX+2)
067: LD (HL),A
068: HIGH-MODE
069:
070: RETURN T$
071: ENDFUNC
```

This user-defined command performs the WEXEC function. I immediately switch to Z80-MODE; get the string control block and subsequently the string address into HL, place the @WEXEC function number into register B, the @WINDOW SVC number into register A, then issue the request. The command then switches back to HIGH-MODE for the return.

```
072: COMMAND WEXEC(M$)
073: Z80-MODE
074: LD IX,&(M$):LD H,(IX+3):LD L,(IX+2)
075: LD B,12:LD A,124:RST 40
076: HIGH-MODE
077: RETURN
078: ENDCOM
```

This is a user-defined multiline function which obtains an argument from the command line. When a program begins execution, the DOS sets up certain program entry conditions. Register HL points to the first non-blank character following the program name. Register BC points to the beginning of the command line. An EnhComp compiled program stores the value of BC into a 16-bit word named "@@BCSV". This is used by one of hte subroutine support librqry functions. Since this label is accessible to only Z80-MODE code, I first switch to that mode to pick up the pointer value then store it in a BASIC variable named "ADDR" and switch back to HIGH-MODE.

Although it would have been far easier (and quicker) for me to perform the command line scanning in assembler, I chose to do it in BASIC because that may be easier for the non-ASM programmer to follow. Recognize, though, how easily it is to switch back and forth between Z80 and BASIC modes.

The REPEAT-UNTIL loop continues to examine command line characters until either a SPACE or RETURN character is found. If a RETURN is found, a null string is returned; this indicates no command line parameter was found. The presence of a SPACE causes the inner REPEAT-UNTIL loop to skip over SPACE characters until the next non-SPACE character is detected. If that has a value greater than or equal to a "0", it is assumed to be a parameter. The routine then sets the test character to a RETURN so the outer REPEAT loop can terminate.

```
079: FUNCTION GETARG$
080: Z80-MODE
081: LD HL,(@@BCSV):LD (&(ADDR)),HL
082: HIGH-MODE
083:     INDEX = 0: T$=""
084: REPEAT
085:     CHAR$ = CHR$(PEEK(ADDR+INDEX))
086:     INC INDEX
087:     IF CHAR$ = " "
088:         REPEAT
089:             CHAR$ = CHR$(PEEK(ADDR+INDEX))
090:             INC INDEX
091:         UNTIL CHAR$ <> " "
092:         IF CHAR$ => "0"
093:             T$=CHAR$
094:         ENDIF
095:     CHAR$=CHR$(13)
096:     ENDIF
097: UNTIL CHAR$=CHR$(13)
098: RETURN T$
099: ENDFUNC
```

This is a straightforward routine to convert an assumed one-character string to uppercase if the value is within the range 'a-z'; otherwise, the string is returned unchanged. It would be easy to adapt this to a generalized routine to convert an n-length string to upper case.

```
100: FUNCTION TOUPPER$(T$)
101: IF T$ < "a" OR T$ > "z"
102:     RETURN T$
103: ELSE
104:     RETURN CHR$(ASC(T$)-32)
105: ENDIF
106: ENDFUNC
```

MISOSYS Products Tidbits

MC C Compiler

MC [DOS 6.x M-21-064] [LDOS 5.x M-20-064]

If you are looking for a **full C compiler**, look no further. If you are looking for a well stocked UNIX System V standard library, look no further. **MC**, reviewed in the January 1987 issue of *80 MICROCOMPUTING*, is a complete C compiler which adheres to the standards established by Kernighan and Ritchie. The library of functions is extensive and System V compatible. The compiler generates Z80 relocatable macro assembler code (M80 or our MRAS). The libraries are files of relocatable object modules. **MC** is a full-featured compiler for the discriminating programmer!

MC supports command line *I/O redirection* for compiled programs, *wild-card* file specifications, parsing for UNIX "." extensions in file specifications, *overlay* support (requires MRAS), a full pre-processor, lots of options, and is designed for the programmer wishing the ultimate in C compilers. The package is supplied with the compiler, pre-processor, an optimizer, assembler macro files, C libraries, a Job Control Language file, the header files, and a 400+ page user manual. MC requires the use of either M-80 or MRAS (available separately), 2 disk drives, and upper/lower case.

'BAD BLOCK' errors in MC

Fm Masa-aki Kitajima: I am enjoying writing programs with PRO-MC. Since I recently installed XLR8er board to my 4P, the 256k RAM disk greatly helped me operating the large PRO-MC compiler system.

Unfortunately I found a problem with PRO-MC. While compiling the source programs and running the compiled /CMD programs, I frequently encountered the 'BAD BLOCK' error. I am confused because the error message sometimes appeared but sometimes not, even compiling the same source programs.

I tried to find the explanation about the 'BAD BLOCK' error in the PRO-MC manual, but I could not find any (I missed?).

I have read the short comment on 'BAD BLOCK' which appeared in the previous issue of TMQ. The comment said the error is a memory allocation error owing to the alloc() (or alike) function misusage. I tried to extensively use the memory allocation functions according to a C language text, but I still encountered the same sort of error.

I am afraid the XlR8er board is not fully compatible with PRO-MC, that is, too fast or like that. Since I am not an experienced C programmer, the most doubtful reason for the error is my illegal coding. But, I cannot understand the unpredictable generation of the error.

My XLR8er is running with its full speed. I found my 4Ps keyboard is fast enough to operate without the help of FIXALL filter.

The Operating system is LS-DOS 6.3. My 4P is equipped with 128k memory.

Fm MISOSYS, Inc: Masa-aki, First, let me explain the 'BAD BLOCK' error. This is something generated by the free() function which is part of the MC library. When you allocate memory space via the alloc() function or its derivatives, it maintains information on the allocation in a data space which is attached to the space allocated. Each block of allocated memory is linked to another by a chain of pointers in this extra data space. Another linked list maintains data on blocks which have been freed.

When you invoke the free() function, it checks on the validity of the block being pointed to in the argument to free(). If it determines it to be correct, then the block is freed. But if it determines that the attached data space is erroneous, it issues the 'BAD BLOCK' error.

If you get such errors during compilation, then it could be a bug in MC. We have found one case where bad source code introduced a BAD BLOCK error during compilation when the compiler's fixup code was bad. That has been fixed and you probably read that in TMQ. On the other hand, frequent, unexplainable, random BAD BLOCK errors during compilation may be caused by memory problems.

You state that you are running the XLR8er without FIXALL/FLT. That may be the culprit. I have recently found that FIXALL is absolutely required in all cases - not because of what H.I.Tech said it was for, but because of what else it does. FIXALL also resets to bank-0 on every interrupt which is required. If you are using Rex Basham's HIBANKS program from TMQ, it also has a minor bug and also does not switch in bank-0 during interrupts. This whole thing is mentioned in TMQ II.iv which was mailed May 27th. There is also a revised HIBANKS on Disk Notes 2.4 which fixes the bug and adds code to switch bank-0 in on interrupts. In the

interim, I suggest you install the FIXALL filter and see if your problems disappear.

Also, were such problems evident before you installed the XLR8er? I wouldn't make a statement that the XLR8er is not compatible with MC. If you are experiencing problems in general, perhaps the XLR8er may not be compatible with your particular Model 4. We have found some cases where it just doesn't work in a particular Model 4. I have no explanation at present. But try to determine if your problems are specific to C or random problems with everything. Does this help?

My current system is: Model 4P with XLR8er, LS-DOS 6.3, FIXBANK or HIBANK (original version appeared in TMQ II,iii) with FIXALL filter. XLR8er is usually running in its full speed. I do not installed any other hi-/low-memory programs (I enclosed the list of the current system status via DEVICE command).

*Beware of **pointers*

Fm Masa-aki Kitajima: Roy, Thank you for your response to my question about PRO-MC's BAD BLOCK errors. I think I have understood what BAD BLOCK error is. Because I was running the XLR8er without FIXALL/FLT, I installed the filter according to your suggestion. Fortunately, the BAD BLOCK error has not appeared since the installation of the filter.

However, I encountered another error message even after I installed the FIXALL filter. When I ran my program which was compiled with PRO-MC, the 'error 07H' appeared. I read the TRSDOS manual's error message section to get the explanation about the 'error 07H', and my understanding about the error was that the error indicates the physical disk damage or that the DOS is incorrectly informed that the disk is damaged by an unknown reason.

I invoked the program in the disk. The program CPR/CCC was successfully compiled with PRO-MC without any error messages. When I ran the program with some of its command line parameters specified, two cases occurred. One was the 'error 07H' in the middle of the execution of CPR/CMD at various positions every time, and the other was a simple hang-up of the system on which the system sometimes rebooted. Such erroneous results were observed whether I used the RAM-DISK or not, and also whether I ran the XLR8er at its full speed or not. When I ran the program without any parameters specified, the program finished normally, however, when any one of the parameters are specified, the program frequently stops with or without the 'error 07H' message.

Would you help me to understand the phenomena? I doubt the reason is in my poor coding, however I cannot understand what kind of bad coding reasons the 'error 07H'.

A short explanation of my program CPR would help you to understand: CPR is a 'C' source print utility. It reads the source file and sends it both to the screen (stdout) and to the printer. CPR has some command line parameters which permits the user to specify 1)lines printed per page, 2)page length, 3)line numbering switch, 4)page numbering switch, 5)left margin, 6)tab expansion. These parameters have their defaults and need not be specified necessarily.

```
/* CPR/CCC --- 'C' source print utility ver 2.0 */
#include      <stdio.h>
#define       SP        0x20
#define       PRINTER   "*PR"
#option       INLIB
static FILE *fpi, *fpo;          /* file buffers for
input and output */
static char buf[81];        /* line input buffer */
static int pagenum=1;       /* page numbering */
static int dummy=0;              /* dummy counter for
looputc() */
/* command line parameters defaults */
static int lines = 60;           /* print lines per
page */
static int page=72;              /* page length */
static int pnum=FALSE;           /* page numbering */
static int lnum=FALSE;           /* line numbering
switch */
static int left=0;               /* left margin */
static int tabsp=4;              /* tab => x spaces */
main(argc,argv)
int argc;
char **argv;
{    int line=1;                 /* screen and
printer-head start line */
     int progline=1;             /* program source
line number */
     int byte,column,tab_stop;
     int i=0;
     char sys_date[9],sys_time[9],title[80];
     /* check command line arguments */
     if (argc<2 || argc>3) {
          prompt();
          _abort("command line format error!");
     }
     else if (argc==3 && *argv[2]!='(')
          _abort("parameters must be in
parentheses!");
     else if (argc==3 && *argv[2]=='(')
          fetchprm(argc,argv);   /* fetch command line
parameters */
     /* check parameter values */
     if (pnum)
          if (page-lines<3)
               _abort("too many lines per page to
print page number!");
     else if (page<lines)
               _abort("too many lines per page!");
     /* file open */
     if ((fpi=fopen(argv[1],"r"))==NULL)
          _abort("source file not found!");
     if ((fpo=fopen(PRINTER,"w"))==NULL)
          exit(1);
/* set sys_date, sys_time, title, and filespec */
     strcpy(title,"Program listing of %s
printed-out %s, %s");
     /* source filespec uppercase */
     while (*(argv[1]+i)) {
          *(argv[1]+i)=toupper(*(argv[1]+i));
          i++;
     }
```

```
    clrscrn();
    lmargin();
    printf(title,argv[1],sysdate(sys_date),systime(s
ys_time));
    fprintf(fpo,title,argv[1],sys_date,sys_time);
    looputc(4,'\n',&line);
/* read file, send it to screen and printer */
    while (fgets(buf,80,fpi)) {
        lmargin();
        if (lnum) {       /* line numbering */
            printf("%3d ",progline);
            fprintf(fpo,"%3d ",progline);
            progline++;
        }
        if (strcmp(buf,"/**/\n")==0) {
            biputs(buf); line++; nextpage(&line);
        }
        else if (strcmp(buf,"/**\n")==0) {
            biputs("/*"); hyphens(); biputc('\n');
line++;
        }
        else if (strcmp(buf,"**/\n")==0) {
            hyphens(); biputs("*/\n"); line++;
        }
        else if (strcmp(buf,"/*-*/\n")==0) {
            biputs("/*"); hyphens();
biputs("*/\n"); line++;
        }
        else {
            byte=column=0;
            while (*(buf+byte)) {
                if (*(buf+byte)=='\t') {
                    tab_stop=column;
                    looputc(tabsp-
(tab_stop%tabsp),SP,&column);
                }
                else { biputc(*(buf+byte));
column++; }
                byte++;
            }
            line++;
        }
        if (line>lines) nextpage(&line);
    }
    line--;       /* strip count for eof line */
    nextpage(&line);
    fclose(fpi); fclose(fpo);
    puts("end of listing");
}
fetchprm(argc,argv)
int argc;
char **argv;
{   int parmc=0;
    int i=0;
    char **parmv;
    char *alloc();
    /* fetch parameters */
    parmv[parmc]=alloc(10);
    while (*(++argv[2])!=')')
        if (*argv[2]==',') {
                *(parmv[parmc]+i)='\0';
                parmc++; i=0;
parmv[parmc]=alloc(10);
        }
        else ( *(parmv[parmc]+i)=*argv[2]; i++; }
    *(parmv[parmc]+i)='\0';
    parmc++;
    /* set parameters */
    while (--parmc>=0)
        switch (*parmv[parmc]) {
            case 'f': case 'F':    setparm(&page,pa
rmv[parmc]); break;
            case 'l': case 'L':    setparm(&left,p
armv[parmc]); break;
            case 'p': case 'P':    setparm(&lines,
parmv[parmc]); break;
            case 's': case 'S':    pnum=TRUE;
break;
            case 't': case 'T':    setparm(&tabsp,
parmv[parmc]); break;
            case 'n': case 'N':    lnum=TRUE;
break;
            default: _abort("parameter error!");
        }
}
setparm(param,arg)
int *param;
char *arg;
{   int i=2;
    /* parameter format test */
    if (*(arg+1)!='=') _abort("parameter format
error!");
    while (*(arg+i))
        if (isdigit(*arg+i)) _abort("illegal
parameter value!");
        else i++;
    /* set parameters */
    *param=atoi(arg+2);
}
looputc(max,c,count)
int max,c,*count;
{   int i;
    for (i=0;i<max;i++) {
        biputc(c); (*count)++;
    )
}
lmargin()
{   looputc(left,SP,&dummy);
}
nextpage(line)
int *line;
{   int j;
    looputc(page-*line-((page-
lines)/2+2),'\n',&dummy);
    if (pnum) {
        looputc(37,SP,&dummy);
        printf("- %d -",pagenum); fprintf(fpo,"- %d
-",pagenum);
    }
    biputs("\n\n");
    for (j=0;j<15;j++) fputs("----+",stdout);
    biputc('\n');
    looputc((page-lines)/2,'\n',&dummy);
    pagenum++; *line=1;
}
hyphens()
{   looputc(75-left,'-',&dummy);
}
biputs(string)
char *string;
{   fputs(string,stdout); fputs(string,fpo);
}
biputc(c)
int c;
{   putchar(c); putc(c,fpo);
}
_abort(string)
char *string;
{   printf("\n%s\n",string);
    exit(0);
}
prompt()
{   int i=0;
    clrscrn();
```

```
    puts("CPR --- C source print utility, ver 2.0");
    puts("Masa-aki Kitajima, Jun. 1988\n");
    puts("command line format : CPR  SOURCE_FILE
(p1,p2,...)");
    puts("parameters :                    default");
    puts("  p=xx; lines printed per page  60");
    puts("  f=xx; page length             72");
    puts("  n;    line numbering switch   off");
    puts("  s;    page numbering switch   off");
    puts("  l=xx; left margin              0");
    puts("  t=xx; tab expansion            4");
    puts("CAUTION : parameters must not be separated
by spaces.\n");
}
clrscrn()
{   fputs("\x1c\x1f",stdout);
}
```

Fm MISOSYS, Inc: Dear Masa-aki, Let me cover the problem with CPR. I believe you have a programming error in the fetchprm() function. You have declared a variable, *char **parmv* which is a pointer to a pointer to char. According to your code, you are breaking up the single string parameter list into a series of strings. Essentially, you are using the *parmv* variable as an array of pointers to chars. That's okay since the two are somewhat synonymous. On the other hand, you go about dynamically allocating space (10 bytes) to the array elements **but the array itself has never been allocated.** You either have to have a statement such as **`parmv = alloc(12)`** or you need to define it as **`char *parmv[6];`**. Here's why your program crashed when you entered parameters. The statement, *`char **parmv`* allocates a 2-byte word on the stack; this word is not initialized and will contain whatever was in that location of memory when the program started. When the parameters were parsed by your function, the un-initialized contents were used as a pointer to the base of the "array". When the pointer to the allocated 10-byte region was assigned to *`parmv[parmc]`*, that could have been anywhere in memory - even in the DOS region. That's why the crash occurred.

Changing LRL with MC

Fm Masa-aki Kitajima: Mr. Soltoff, I am a LS-DOS (ID# 16939) /PRO-MC (ser.# 00236) user and am enjoying developing utility programs with PRO-MC. Its well conserved compatibility to K&R greatly helps me to recompile and run the same C programs on the MS-DOS machines which are popular in my company's laboratory. Although PRO-MC system is rather big for my 4P's pretty [small] main memory, the XLR8er's 256k RAM disk provides a wonderful environment in which I can do source-code writing (with SAID) /compile/link all in the memory. It's amazingly fast. I thank you for the excellent C compiler software.

By the way, I have a question about the logical-record-length (LRL) of the disk files generated by the C programs which were compiled by PRO-MC. When I was developing a C program which generates DOS-help-text-files, I found I have to make the LRL of the help text files to one. All the MC-generated files I examined had the LRL of 256. Unfortunately I failed to find a way to create files of LRL of 1 with PRO-MC, though I know it's simple from BASIC.

The Technical Reference Manual of the DOS says that the LRL of files can be specified or altered only by CREATE or COPY commands. Can I control the LRL of the files from C programs? I understand that C's character I/O functions provide an effect as if I were accessing the files of LRL of 1, however, it does not help me in this case because DOS HELP command handles the help text files which LRL is actually one.

I expect your advice. Thank you for reading.

Fm MISOSYS, Inc: The MC stream and block file I/O functions will always use an LRL of 256. It will ignore LRL Fault errors on opens and will always create a file with an LRL of 256. If you need to create a file with a record length of other than 256 (to be used for some external processing), than you can easily change the LRL in the directory by using the *`gattrib()`* and *`sattrib()`* functions which are part of the MC 1.6 upgrade. That pair of functions can also be used to change flag attributes, and password(s).

EnhComp BASIC compiler

EnhComp [DOS 6 M-21-072] [LDOS 5 M-20-072]

This is an enhanced **BASIC compiler** released in 1986 and reviewed in the March 1987 issue of *80 Microcomputing* and October 1987 issue of *COMPUTER SHOPPER*. This compiler has lots of great features. It handles the bulk of Model III Microsoft BASIC and supports additional commands and functions. Standard is floating point with both **single and double precision** functions; random file access ("X" mode for reclens to 32767), turtle graphics, pixel graphics, keyed array sort, multi-lined functions, user commands, IF-THEN-ELSE, REPEAT-UNTIL, printer control, sequential file positioning, line labels and more.

A *supervisor* program automates the edit-compile-test phases inherent when using compilers; this makes using EnhComp almost as easy to use as your BASIC interpreter. You also get CED, a **line editor** with string search/change, partial load/save, renumber, copy, and move.

Enhcomp has a built-in **Z80 assembler.** You can **easily** create hybrid programs of BASIC statements and in-line assembly code which **completely eliminate** contorted string packing and DATA statement high-memory module techniques for your BASIC program to access a machine code module. Z80-MODE accesses BASIC's variables!

You'll have to edit *existing* BASIC programs, but the power and completeness of EnhComp make that an easy task.

Bug in error handling

Fm MISOSYS, Inc: As reported elsewhere in this issue, I experienced some problems with EnhComp trapping some of my programming errors while I was developing the PAW/BAS program. For instance, I had neglected to put a DEFINT A-Z statement in my program; thus, variables without suffix modifiers defaulted to single precision type. The INC and DEC operations are restricted to integer variables. EnhComp correctly trapped the error; however, the due to a programming error in the compiler, the error reporting caused a crash. I fixed that up with a 2-byte patch.

In another example, I left off one double quote from a structured IF statement's expression. This invalidated the IF causing the following ELSE to be treated as an "ELSE without IF" error. The compiler's error handling procedure was also incorrect in that it was subject to a program stack problem which caused a crash. A one byte patch fixed that up.

Lastly, due to a bug in the WAM22/FIX which didn't properly handle the case of a application being invoked external to PRO-WAM activation, I found one case of the BC compiler not closing up the BASIC source file under an error condition. That too was fixed up by a patch; the Model III mode version doesn't require that fix up. Here's the two patches: BC54/FIX for the Model III mode EnhComp and BC64/FIX for the Model 4 mode PRO-EnhComp.

```
. BC54/FIX - 07/25/88
. Patch to EnhComp's BC/CMD
. Patch fixes up error handling on
. INC & DEC type mismatch errors
. and ELSE without IF error.
. Apply via, PATCH BC BC54
D2B,61=03 67:F2B,61=97 07
D2B,81=03 67:F2B,81=97 07
D2D,3A=CC:F2D,3A=CA
. Eop
```

```
. BC64/FIX - 07/25/88
. Patch to PRO-EnhComp's BC/CMD
. Patch fixes up error handling on
. INC & DEC type mismatch errors
. and ELSE without IF error. Also corrects
. closing of source file after pass 2.
. Apply via, PATCH BC BC64
D05,88=AF CD 42 31 3A 2B 88 C9
F05,88=00 00 00 00 00 00 00 00
D22,AD=CD 70 2F;F22,AD=3A,2B 88
D2B,70=15 3B;F2B,70=AC 07
D2B,90=15 3B;F2B,90=AC 07
D2D,49=CC;F2D,49=CA
. Eop
```

LB Database

Little Brother-M4 L-50-510

LB is a **flat file data management system** where ease of use is its primary goal; you don't need to program anything or remember complicated command sequences to manage data. Even for the most complex data management needs, LB produces results quickly; EVERY function in LB is menu driven and comes with complete on-line HELP information.

To set up a data base, you just define the record layout. For each field, enter a descriptive name, type, and length. LB handles up to **65534 records**; each can contain up to **1024 characters.** LB supports up to **64 fields per record**; fields may be up to 254 characters long. There are seven types of data fields available: alpha, numeric, right justified, literal, dollar, float, and calculated (add, sub, mul, and div); any of which may be a *Protected Field*, so that its data will not be displayed unless the proper *Password* is entered.

You next establish a *screen*, and you are ready to begin entering data! You may view or edit any record at any time. Find information quickly. You can even create an *index* to your data so any record can be accessed within seconds.

Simply define a print format screen, and LB will print records according to your specifications; 10 different formats can be created. You can print with headers/footers, date, time, page numbering, totals and sub-totals if desired, mailing labels format, and even form letters. You select what records get printed and can use an index for printing in *sorted* order as well; great for organizing your report.

For **automating your processing**, LB can be run in an *automatic* mode; frequently used procedures (such as selecting, sorting and printing) can be saved for future use.

LB requires a minimum of two floppy disk drives and 128K of RAM or hard disk, 64K, and one floppy disk drive).

LB's startup drive access

Fm Tony Sowers: There's a lot of drive access that goes on during loading of LB. I have several drives and, even with

no disks in the drives, the lights come on multiple times eating a noticeable amount of time.

Since a simple "LB" loads with no data files, it would seem that the actions are unnecessary (seem, I said).

Fm MISOSYS, Inc: Yes, Tony, I would suspect that you would feel the multiple drive search would be unnecessary when you don't even enter a data base name on the command line. However, due to LB's operating environment, these accesses are necessary. Let me explain what happens.

LB operates as a set of overlays. There are one or more overlay modules which support each menu command. There's also a root or main module which oversees the menu processing. Communication among the various overlays is achieved via an environment data file which is created and updated at various stages of LB processing.

When you invoke "LB", the following sequence of events occurs:

1. The LB/CMD file is searched for and loaded by the DOS. This causes a drive search for LB/CMD. Under MS-DOS, that would be the LB.EXE file. MS-DOS would search the current directory and any PATH designated in the environment. LS-DOS searches all drives.

2. LB invokes and creates the environment file, LB/ENV. Under LS-DOS, the @INIT service call used to create or open a file must first search all drives to see if the file exists. There's another drive search. Under MS-DOS, the file would be created in the current directory. The environment file is then written with data which is later used to indicate a start up activity; that LB has been invoked from DOS. The file is then closed.

3. The LB0 overlay is loaded and invoked. Under LS-DOS, this causes a drive search. The primary LB processing loop is executed. This loop takes control after the completion of each LB overlay. Thus, when you complete a print, or edit/update, or screen definition, etc., the LB0 module takes control and re-executes this loop. Assuming that the current mode is the initial startup of LB, the following takes place in this loop:

a) The LB environment file is opened. Again this causes a drive search.

b) The first 16-bit word is read. This contains a code which indicates the last overlay which was executing.

c) A new entry code indicating LB0 is written to the environment file and then the file is closed.

d) Recognizing that the LB0 entry was made from the startup module, the command line is checked for parameters such as data base name and password.

So we have a number of processes at the startup of LB which cause a global drive search. At least one of those search procedures is guaranteed to examine all drives - that's the initial creation of the environment file - unless one is hanging around from some earlier LB invocation which was prematurely terminated by some external influence (power failure, reboot, etc.).

Now one way to speed up the process would be to force LB to always use a particular drive for the environment file. I think that is too severe a restriction. I recollect such restrictions in some commercial software packages which hard coded a drive specification. Perhaps in the next release of LB, I could make that an option.

PRO-WAM

PRO-WAM 2.0 M-51-025

This desktop manager gives keystroke access to 4 memory resident **pop-up** applications and disk access of others. A Function Key lets you invoke DOS library commands. PRO-WAM tums your TRS-80 into a powerful machine because it comes with many useful and powerful time savers and desk organizers. Here's some of what you get:

4 An **ADDRESS** file data base prints cards and mailing labels. Throw away that black book and your Rolodex file.

4 **HEAD** pipes formatted address data into your letters.

4 **BRINGUP** tickler file schedules up to 12 items per day by time. New print module. Remember those appointments.

4 **CAL**endar gives you a month at a glance; covers 4000 years. Flags days with BRINGUP items.

4 A 3x5 **CARD** filer for a free-form scratch pad of 40 columns by 12 rows. Or use the new **CARDX** with ***forms*** capabilities. It's great for small data base.

4 **PHRASE** is a KSM from disk for lots of automation.

4 A telephone list and auto**DIALER** for Hayes modems.

4 **CAL**Culator gives you 4-functions at your fingertips. **RPNCALC** gives 7-functions in bin, oct, dec, and hex.

PSORT puts your PRO-WAM data files in sort order. *EXPORT* and *IMPORT* functions allow you to move data across windows between applications and programs. There's even an online **HELP** facility!

PRO-WAM works with all programs which use standard DOS keyboard requests and honor the DOS high memory pointer; requires one 32K RAM bank, about 2K of high memory, and a small piece of low RAM. If you have a model 4, then you must have PRO-WAM!

@WEXEC bug in WAM22/FIX

While I was engineering and developing the PAW utility for this issue of TMQ, I happened upon a problem with the code

introduced by the WAM22/FIX. Recall from TMQ II.iv that the WAM22 fix was designed to allow PRO-WAM to operate properly even if the video display was in 40 character per line mode when PRO-WAM was activated. The fix worked correctly to deal with that problem. I also don't think anyone who applied that fix got into any trouble. But let me elaborate.

The PRO-WAM high-memory module is a very complex piece of code. There really are two distinct processes at play there and two distinct entry points. One process is the keyboard device filter which gets entered on every @KBD service call. That means when ever any program or the DOS is asking for a check of the keyboard. The other is the @WINDOW service call which gets entered by those programs requesting WINDOW services.

The PRO-WAM release 2 interface added further complexity with the addition of the @WEXEC facility which supports the invocation of a PRO-WAM application under program control rather than explicit keyboard depression. That's the facility I used in the CAL to BRINGUP interface.

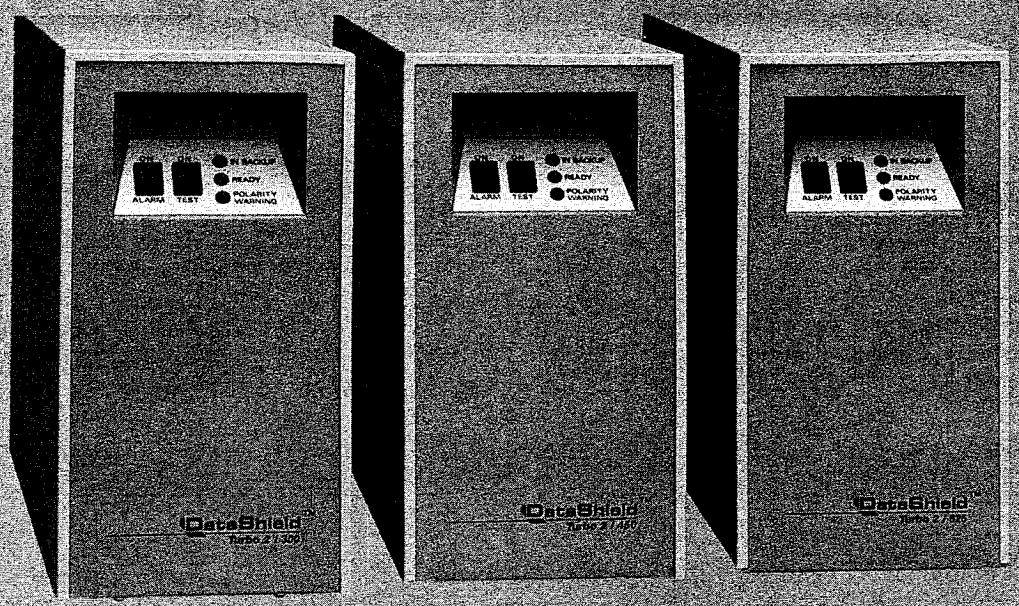
Now when PRO-WAM first activates with the <CTRL-P> request, it has to do some significant setup. It first modifies some resident code so further filtering of keystrokes bypass the startup code. It then passes control to code which processes entry into the PRO-WAM environment. That code needs to determine if PRO-WAM has just been activated and no window is open (recursion level of 0). Under that circumstance, other setup code needs to be executed. The program stack has to be switched to the high memory stack, the network flag has to be saved and the file-open bit set to network operation, and the number of the current DOS overlay needs to be saved. I also had previously coded the management of the video mode to this process with the WAM22 fix. When these things are done, an environment flag is set. There are also some things which need to be done prior to and subsequent to the execution of a PRO-WAM application. One is to save the current state of inverse video.

When an application completes, the window recursion level is tested. If no windows remain open (recursion level of zero), additional cleanup duties must take place. Since it is possible that a WINDOW service request could be made without activation of PRO-WAM, the exact nature of processes performed depends on the status of the environment flag.

If an application is invoked via the @WEXEC function without activation of PRO-WAM, the environment switch would not be made; unfortunately, the WAM22 fix added the code which restores the video mode to the global cleanup process rather than the cleanup process which is executed just when the environment flag is set. The problem was that my PAW program was using @WEXEC external to PRO-WAM activation, the video mode was not captured, but it was "restored". Since the video mode is stored in the DOS MFLAG$, which also keeps a flag for enabling the external I/O port, whenever I exited an application invoked via PAW, the external I/O port was disabled; this action disabled my hard drive. Since the hard drive was disabled, the next attempt to invoke an application resulted in a locked up system. The hard disk driver was caught in a never ending loop waiting for the hard drive to get ready. The solution to all of this was to revise the video mode handling routine to not do any cleanup unless the environment flag was set. The small patch, WAM23/FIX, which follows corrects for this behavior.

```
. WAM23/FIX - 07/11/88 - Patch to PROWAM
. Revises set to 80CPL mode when
. activating PRO-WAM
. as patched with WAM22/FIX
D01,61=73 47;F01,61=22 3D
D0A,DB=53;F0A,DB=72
D0A,E9=CD 72 47;F0A,E9=32 00 00
D14,42=32 00 00;F14,42=CD 53 47
D16,4C=22 3D;F16,4C=73 47
. Eop
```

NEW SECOND GENERATION SURGE SUPPRESSORS



FEATURES

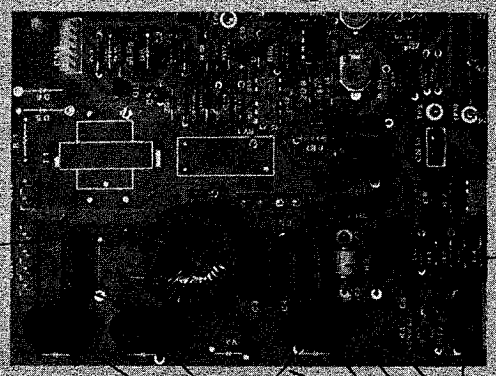
- Under 1 ms Transfer Time
- Regulated Output
- Electronic Overload Protection
- Self Test
- Full Battery Protection
- Sight Wiring Diagnostics for Reverse Polarity
- Rapid Battery Recharge
- Audible and Visual Alarms
- Phase lock to AC line return
- Full Surge and Noise Protection
- Remote Signal interface-indicates when system is on Battery or when Battery is low
- Shipped Battery Complete
- Average Load Back-up Time 15 minutes
- 4 NEMA 5-15 R outlets

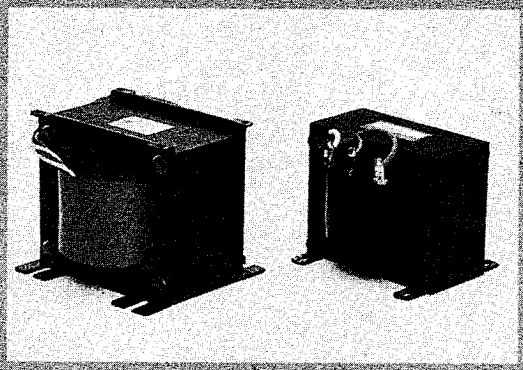
SPECIFICATIONS

Turbo 2 Models	300,450,625
Back up time (average load)	15-25 minutes
Output Rating	300VA, 450VA, 625VA
Transfer Point	104VAC
Input/Output Frequency	60 Hz
Typical Recharge Time	4-5 hrs
Transfer Time	Less than 1 MS
Wave Form	Psuedo Sine
Low Voltage Alarm	Buzzer & Light
Overload Protection	Electronic, Shuts down
Weight pounds	23,25,27
Kilograms	10.4,11.3,12.2
Size-Inches	5.9x11.5x18.6 in.
Centimeters	15.0x29.5x47.2
Surge Protection Energy absorption	200 Joules
Power Dissipation	1,000,000 watts (100 microseconds)
Noise Protection Mode	Transverse & Common
Attenuation	-85db to -20db
Frequency	100 KHz to 100 MHz
Clamping Voltage Level Peak	200 VAC

SECOND GENERATION SPS TECHNOLOGY

THE TURBO 2/300, 450 AND 625 OFFER THE FOLLOWING FEATURES

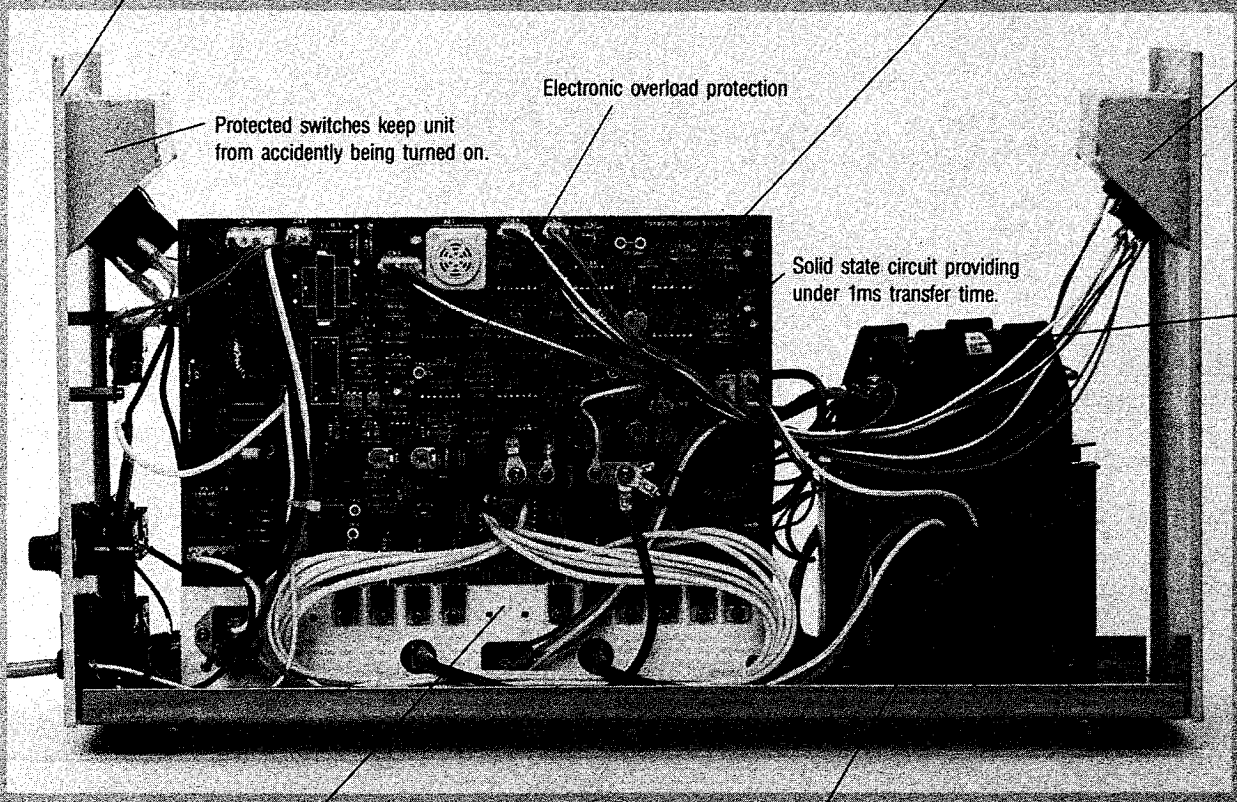




Surge and Noise Protection:
Turbo 2 fully protects all incoming legs (both normal and common mode) from surges, spikes, and line noise. Three MOVs, five capacitors, a toroidal inductor and gas discharge tube provide 200 joules of surge protection and up to 85db of line noise attenuation.

Oversized Transformer:
The very heart of a standby power system is the transformer. The unit output power and load handling is directly related to the transformer size, quality of material and workmanship. Our high quality heavy gauge transformer is 27 percent larger than the typical competitors transformer for the same VA power rating.

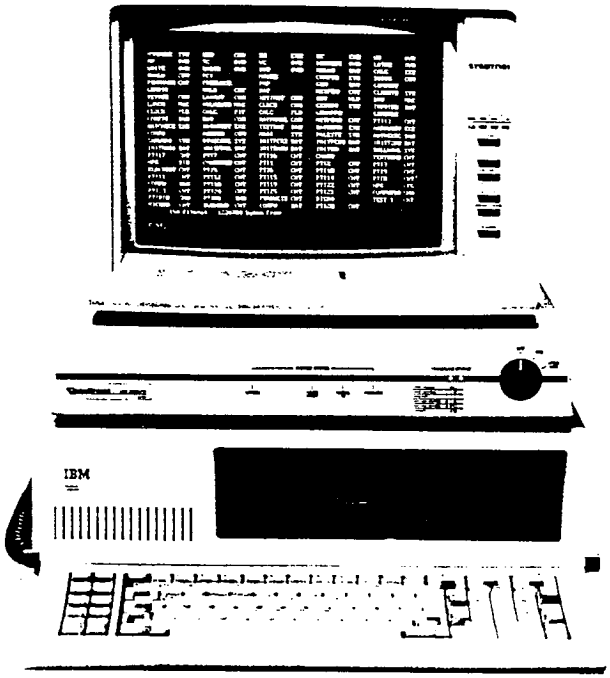
ADVANCED TECHNOLOGY



©1992 PTI Industries

PTI continues its technological leadership in the introduction of a new concept in UPS design. Previous design criteria required large, costly transformers. Our engineers have reduced the size, weight, and expense with this revolutionary solid-state design.

The benefit to you is an affordable UPS that does not take a fork-lift to move. The solid-state design and quality engineering insures a level of reliability not achievable in other UPS products.



Compare the SS400 and SS700 to other UPS currently on the market.

- Compare price
- Compare size
- Compare weight
- Compare quality engineering

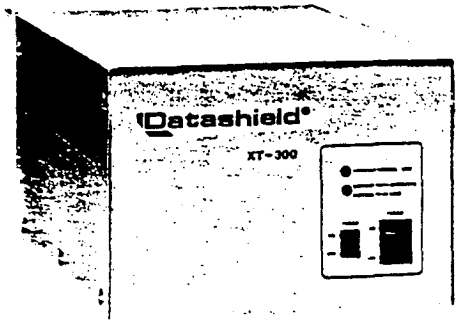
This new SS Series is in a class by itself.

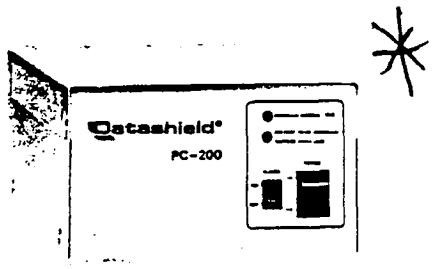
DATASHIELD SS400 AND SS700 SPECIFICATIONS

INPUT VOLTAGE:	135 VAC to 90 VAC
INPUT FREQUENCY:	60 Hz
SURGE PROTECTION:	200 Joules
INRUSH CURRENT:	300% of maximum output utilizes "Soft Start" technique to bring up fully configured IBM-AT
OUTPUT VOLTAGE:	120 VAC ±3%, regulated
OUTPUT POWER:	SS700, 700VA; SS400, 400VA
OUTPUT FREQUENCY:	60 Hz ±0.1, Crystal controlled
OVERLOAD PROTECTION:	Current limiting electronics
TRANSFER TIME:	**NONE**
OPERATING EFFICIENCY:	Greater than 90%
NOISE ATTENUATION:	Greater than 80 dB normal mode — Greater than 40 db common mode
AUDIBLE NOISE:	Less than 50 dBA
BATTERY:	Industrial grade sealed lead acid
BACKUP TIME:	Minimum 5 minutes full load — 20 minutes average load
RECHARGE TIME:	One hour to 90% charge — Two hours to 100% charge
STATUS INDICATORS:	Normal AC (AC line less than 104V) Overload (excessive output load)
	Power Boost (low line supplemented) Battery (battery condition-LED & audible alarm)
OPERATING TEMPERATURE:	0°c - 40°c
OPERATING HUMIDITY:	95% maximum relative humidity without condensation @ 25°c
WEIGHT:	Model SS700 39 lbs — Model SS400 32 lbs
SIZE:	Height 3" x width 18" x depth 17"

DataShield

PTI Industries • 269 Mt. Hermon Rd., Scotts Valley, CA 95066 • Telex 176841 • Fax (408) 438-0967
A Pentron Company





XT300
Designed to work with all personal computers with internal or external hard disks. Provides 300 watts of power. Four plugs provide for any equipment that needs protection.

PC200
Designed to work with all personal computers that are floppy disk based such as Apple II, Apple III, Apple Macintosh, and IBM PC. It provides 200 watts of power.

Technical Data/Back-Up Power Sources

SPECIFICATION—MODEL	AT1500	AT1200	AT 800	AT500	Turbo 350	XT300	PC200
Back-up Time (Average Load)	15 to 25 minutes	15 to 25 minutes	15 to 25 minutes	15 to 25 minutes	15 to 25 minutes	15 to 25 minutes	15 to 25 minutes
Output Rating	1500 Watts	1200 Watts	800 Watts	500 Watts	350 Watts	300 Watts	200 Watts
Transfer Points:							
Nominal 120 VAC	104 VAC	104 VAC	104 VAC	104 VAC	104 VAC	104 VAC	104 VAC
Nominal 220 VAC (1)	187 VAC	187 VAC	187 VAC	187 VAC	187 VAC	187 VAC	187 VAC
Nominal 240 VAC (1)	204 VAC	204 VAC	204 VAC	204 VAC	204 VAC	204 VAC	204 VAC
Input/Output Frequency							
120 VAC Units	60 HZ	60 HZ	60 HZ	60 HZ	60 HZ	60 HZ	60 HZ
220/240 VAC Units (1)	50 HZ	50 HZ	50 HZ	50 HZ	50 HZ	50 HZ	50 HZ
Typical Recharge Time	10-12 Hours	10-12 Hours	10-12 Hours	10-12 Hours	10-12 Hours	10-12 Hours	10-12 Hours
Transfer Time:							
Typical:	4 Millisec	4 Millisec	4 Millisec	4 Millisec	See Note 2	1 Millisec	4 Millisec
Maximum	10 Millisec	10 Millisec	10 Millisec	10 Millisec		4 Millisec	10 Millisec
Wave Form	Sine	Sine	Sine	Sine	Pseudo Sine	Pseudo Sine	Pseudo Sine
Low Voltage Alarm	Buzzer & Light	Buzzer & Light	Buzzer & Light	Buzzer & Light	Buzzer & Light	Buzzer & Light	Buzzer & Light
Overload Protection	Current Limiting Inverter Circuit and breaker	Current Limiting Inverter Circuit and breaker	Current Limiting Inverter Circuit and 10 amp fuse	Current Limiting Inverter Circuit and 10 amp fuse	Shuts Down	4 AMP Fuse	3 AMP Fuse
Weight-Pounds	139.8	133.2	72	58.5	39.5	37.5	24
Kilograms	63.4	60.4	32.6	26.5	17.9	17.0	10.9
Size-Inches	12.6x9.3x16.5	12.6x9.3x16.5	12x8 5/8x15 1/2	12x8 5/8x15 1/2	5.5x11.5x18.6	7 7/8x10 3/8x13 1/2	7 1/4x4 1/4x13 1/2
Centimeters	32.0x23.5x42.0	32.0x23.5x42.0	30.5x22.1x39.4	30.9x22.1x39.4	14.0x29.3x47.2	19.9x26.3x34.3	18.4x10.8x34.3
Surge and Noise Protection:							
Power Dissipation (100 Microseconds):	1,000,000 Watts	1,000,000 Watts	1,000,000 Watts	1,000,000 Watts	1,000,000 Watts	1,000,000 Watts	1,000,000 Watts
Energy Dissipation	100 Joules	100 Joules	100 Joules	100 Joules	100 Joules	100 Joules	100 Joules
Mode Noise Protection:	Transverse & Common	Transverse & Common	Transverse & Common	Transverse & Common	Transverse & Common	Transverse & Common	Transverse & Common
Attenuation:	-40db Minimum	-40db Minimum	-40db Minimum	-40db Minimum	-20db Minimum	-20db Minimum	-20db Minimum
Frequency:	100KHZ to 300MHZ	100KHZ to 300MHZ	100KHZ to 300MHZ	100KHZ to 300MHZ	100KHZ to 300MHZ	100KHZ to 300MHZ	100KHZ to 300MHZ
Clamping Voltage Level Peak	200V (120V); 370V (220/240V)	200V (120V); 370V (220/240V)	200V (120V); 370V (220/240V)	200V (120V); 370V (220/240V)	200V (120V); 370V (220/240V)	200V (120V); 370V (220/240V)	200V (120V); 370V (220/240V)

NOTE 1: 220/240 VAC Units end with suffix E., e.g.:AT800E
NOTE 2: Under 1Ms; Zenith Emulation

The Hardware Corner

International Keyboards

Fm Bob Hyde: Yes Lynn Sherman, there are French keyboards, and Lescript caters for them. One types % after the command and all is well. (well nearly well, the dictionary is a bit bizarre. I never knew my English spelling was so bad!). The keyboard takes some using, I sometimes transpose on the fly, as you say.

Chorus: For A is a Q and Q is an A, We do things the AZERTY way.

As for the directory we have funny words such as: "FOR,QT", "RENQ,E", ",E,DISK", ",E,ORY", "HERTW/JCL". Hoz zould you like to hqve to zrite like this? A,ericqn.

Finally, I do think that a 3000 Franc hardisk system is a bargain. I have started to save up.

Le TRSDOS 6.1.1 F est un systeme d'Exploitation de Disques a 8 bit qui vous permet d'utiliser votre Modele 4p avec un clavier AZERTY. If you need the addendum to the manuel for 'version internationale' just say. It is in French.

Fm Fred Pieters: Dear Mr. Soltoff, First of all, I would like to thank you for sending my letter to Mr. Michel Houde, Compiegne, France, who changed my LDOS 5.3 for my French-keyboard Model IV. I also would like to order LS-HOST/TERM L-35-201 for handling my Kantronics KAM Radio Modem with my Model IV in Mod.4 mode. Kantfonics has no terminal program available unless Pacterm, which works in the Model III mode. The ordering card is included. The only condition is that the Host-term program uses the DOS keyboard and video drivers and not its own.

I am always delighted to receive your excellent TMQ, but receiving Vol.II.iv, I was rather puzzled by the fact that no subscription form for the new Vol III was included. Is TMQ discontinued? If not I want to subscribe to this next Volume.

In different volumes of TMQ I read about the Keyboard confusion in the minds of different US users of the TRS8O computers. I'll try to enlighten your readers on this matter, if you are interested. The Model I had only one version: the American Keyboard, which we will call the QWERTY keyboard.

The Models III and 4 are available in Europe with three keyboards: the QWERTY, AZERTY (French) and the QWERTZU (German) Keyboards. The names are, as you have noticed, related to top row of the letters. Tandy changed the space-compression codes for the international character set. Included are the Model 4 Keyboard layout and the Model III AZERTY ASCII Character set. Of course the keyboard drivers of LDOS and TRSDOS differ from the QWERTY DOS. It is therefore that some programs, who use their own keyboard driver, like SuperUtility + and Powertool, or excellent wordprocessors, cannot be used and we are confined to the bugged Superscripsit, who loses text after five pages! There is no Belgian Keyboard, as there is no Belgian language. In Belgium, (about ten million inhabitants), people in the northern part of the country are called Flemings and speak Dutch (like in the Netherlands), people in the southern part of the country are called Walloons and speak French and in a small eastern part of the country German is spoken.

LSDOS 6.3 has been released by Tandy in a French and a German version in October 1987, but Tandy saw no point in making a LDOS 5.3 French and/or German Version. I can assure you that the job Mr. M. Houde did on the DOS is excellent. If you are interested, you could always ask him for his keyboard driver. I saw no harm in sending you a copy of the Tandy LSDOS 6.3 version, and I would be very interested if you had any comments on the matter or if you would find any bugs and report on them.

Well, I think this clarifies the exotic keyboard matter. I am available if you would want more information on this matter.

Fm MISOSYS, Inc: Thank you, Fred, for the lesson on keyboards. Actually, although I was aware of the three international version keyboards from my early involvement with the LDOS release which Tandy sold, I really never knew them by the terms "AZERTY", or "QWERTZU". Gosh, the latter sounds like the name of some alien creature or robot reminiscent of Gort in *The Day the Earth Stood Still*. "QWERTZU, klattu borado nicto!"

I've heard of "Flemings", but I never heard of "Waloons". To tell you the truth, I'm not so sure I would want to be called a "Waloon". Maybe it loses something in the translation to English.

When I next get the chance (I'm only about two months behind in most things), I'll see if that international 6.3 has all

of the LSI patches to 6.3 installed. Also, I would be able to uncover the actual keyboard matrix by examining the keyboard driver. According to my old information on the Model III international keyboards, the extra keys are wired to matrix row four (out of eight) and columns four through eight. That's the row of the matrix where the American keyboard only has three keys: X, Y, and Z. In looking at a facimile of the German keyboard, I certainly can see how one can refer to it as "QWERTZU". What I don't have is the keyboard matrix layouts for the Model 4 international version keyboards. But I suspect that I could take the row eight additions from the American Model 4 keyboard and add that to the Model III international matrix versions. Sound right?

Hard Drives

Questions you never asked...

Fm Charles A. Ainsworth: Roy, Your recent flyer and the latest TMQ ask for feedback in response to your proposal of providing 20 meg hard drives with the complete works.

Before I get into the subject, let me clarify that I am not writing this letter expecting any specific reply but rather as my contribution in assisting you to consolidate my ideas with whatever others you may receive, perhaps as an aid to all concerned in reaching a decision on the purchase of your drives.

I have many questions on hard drives that no one I could really trust has ever answered. I have ruled out magazine articles and reviews as they obviously would not criticize hard drives too energetically as most magazines are, unfortunately, too beholden to their advertisers.

I have several friends, some of which are model 4 users and others MSDOSers. I have posed some of my questions to them and find they don't know many of the answers themselves even though some are hard drive users. So this letter might be called a composite of questions and uncertainties regarding hard drives.

As things are today, and apart from your entry into the HD field, I am scared of hard drives. Maybe you could at some point set my mind at rest (possibly with a general-coverage writeup in TMQ). Here are mg points:

1. One hears of the possibility of head crashes. If you care to look at the Tandy 1988 computer catalog, page 13, where a 20 meg disk cartridge system is offered, you will see that it mentions "This eliminates head crashes..."

2. If the odd bits of info my friends and I have scraped together from various sources are correct, hard drives spin at 3600 RPM with an air cushion separating the head from the disk. As an engineer with many years experience in many abstruse technologies, I immediately think that 3600 is one heck of a speed to run a disk drive. What about bearings? Are they reliable enough to keep disks properly aligned and balanced at that speed?

3. What happens when the power company pulls the switch on one or when there's a sudden voltage dip? Does the head come down on the disk at, say, 1800 RPM when the air cushion dies, but when the disk is still spinning fast enough to create chaos by head/disk friction?

4. Is an uninterruptable power supply a must to prevent disaster from power failure and resultant crashes? If so, that must be added to HD cost.

5. What happens when there is a crash? How much do repairs cost (if the drive is repairable). To pay several hundred dollars for a drive and have it crash creating a repair bill of another several hundred, isn't my idea of fun!

6. What normal service do hard drives require, if any, and how often? Where is it available and what, Just as a ball park figure, does it cost on an average.

7. If I were to go for hard drives, I guess I would want a pair. Presumably the disk controller, host adaptor, drive case, device drivers and so on, would not have to be duplicated with a second drive, so one would need your $495 (target price) setup plug only the additional bare drive (just as a shot in the dark, say some $300).

8. Are there any typical figures available for the gain in speed for: a) Locating a specific file on a drive and b) Reading/writing a specific file.

9. Would any "surgery" have to be performed on an existing computer to connect the drives (I would assume not) or could the hard drive cable simply be plugged into the card edge connector on the computer in the same way as external floppies (I would assume yes).

10. If I connected two HD's, how many floppy drives could I have on line at the same time?

11. Most of my work demands absolute reliability (or as close to it as I can get) in writing to disk and reading back. On some of my large data bases, unflagged I/O errors can be very serious. When operating with my present floppies, any time data is sent to disk, it is sent straight from RAM to two disks. I always consider the possibility of flaws in disk coating, in which one or more cylinders just got by when formatting but which, might be marginal. So if I write to one disk and unknowingly hit a marginal track, then later back up that floppy to another, possibly the marginal information also goes onto the backup. If I used HD's, I would probably use a similar system by writing to disk in duplicate direct from RAM.

However, from what little I know about it, there must be times when it becomes necessary, for one reason or another, to backup a HD to floppies. What sort of job is that, what would be typical times for backing up and then for restoring data from floppies to HD?

12. I have my present four floppy drives all outside the computer (I don't use the internal drives native to the computer); at times, either due to computer failure or any other reason, it may be necessary to disconnect the drives and move them over to another computer. With my setup, all that requires is unplugging from one machine and plugging into the other and away I go. Would it be as easy as that with HD's? I suspect certain special precautions have to be taken when moving HD'S to avoid head thrashing.

Roy, hard drives have, for years, tempted me and I could well use them. But then, the fear of hardware reliability problems has always scared the socks off me and made me chicken out every time. and repeating what I said above, neither I or my friends have ever found an answer to these questions that could be trusted. In addition, TMQ has also scared me off with all those endless problems with drivers from firms like Aerocomp, Montezuma et al where purchasers went through very hard times getting it all lined up; in all probability, there would be no such thing with your kit, but it would be nice to have it confirmed.

But I do trust you [sounds in Misosys office of shirt buttons popping off due to swelling with pride!! <grin>]. So I feel it would be interesting if you addressed some of these matters. If you satisfied me on them, you could probably count on me for a pair of drives, with all the works, for somewhere around the last Quarter of 1988 or the first of 1989.

Fm MISOSYS, Inc: Charles, The questions you raised are all good ones. Rather than wait until the next TMQ to see the answers in print, let me address them now and re-use this letter later. I'll address your questions point by point to make it easier to follow.

The Tandy ad which references a 20 Meg cartridge drive which "eliminates head crashes" is referencing the Bernoulli flexible cartridge drive. This is essentially a precision floppy drive which uses a Bernoulli-effect read/write head rather than one which touches the surface. The Bernoulli effect is entrenched in a column of air which maintains the head some small finite distance from the media. The result is similar to a Winchester hard drive which uses the "wings" on the head to maintain the "flying" distance above the media. Whether or not the Bernoulli drives are "safer" than hard drives is not the issue. For some years now, I have found hard drives sufficiently reliable to eliminate any personal fear over their use. I have used many drives for many years and have never experienced any difficulty whatsoever once the drive was working to begin with. You are more apt to find a maintenance problem with older controllers than the drive - given an adequately designed environment.

Hard drives do spin at 3600 RPM. Why should we consider that excessive? Do you know the rotational speed of a dentists air turbine drill? You trust your life to an airplane jet engine which has considerable rotational speed. Believe me, there is nothing to worry about that speed of rotation. A hard drive is a precision instrument. If you left a hard drive running continuously, the bearings would probably last 3-5 years. My hunch is that they would be the first to go. Usually, the drive will start getting a little noisy; that's how you can tell the bearings are starting to go. Of course, I have yet to have that happen. I guess I have never had any one drive operating continuously for that long.

I also have had a number of machines in use when the power fails. What's the difference between you turning off your power switch and the power company having a mishap? Same difference. If you are really concerned about that kind of failure, you are a prime candidate for an Uninterruptable Power Supply (UPS). I also think the head will be still flying at 1800 RPM. I'm not sure at what speed it "takes off", but that's not really a big worry. Another means of counteracting that is to use one of the very fast access drives which use a voice coil for head movement instead of a stepper motor. When the power is cut, the voice coil retracts and positions the head to the landing area. Some drives also use a mechanical arm to ensure that the head never touches the surface. These special techniques are prime contenders for portable computers which are subject to external movement. Now an UPS is more important when you want to ensure that the program you are running must not be interrupted. I'm more concerned with that than any worry over the hard drive. And yes, I still don't have an UPS installed here yet. [Take that back, I now have two 450VA UPS systems installed; one on Brenda's AT compatible and one supporting my PC and 4P.]

Here's my opinion about repairing a drive which has malfunctioned. To begin with, when a drive "breaks", you cannot automatically attribute it to a crash. Sure, if you can afford to attempt a re-format and all tracks on a single surface are bad, that probably is a head crash. It's probably simpler and cheaper to just throw the drive away and replace it rather than fixing it. Far too much today is more costly to repair than to fix. Also when you attempt to repair, you still wind up with the un-repaired portion being old and subject to breakage earlier than a new unit. The worry about a hard drive breaking is minimized with adequate protection of the files which are stored on it.

The normal maintenance on a hard drive package is to clean the muffin fan's air filter if it has one. That's it. There is no maintenance.

Some folks do consider the use of a second hard drive as a backup medium for the first. I've never used that method. I've always used floppies. I did consider a tape drive once; in fact, I even have a Cipher FloppyTape drive which I was going to write drivers for. That works off of a floppy controller. But I never got around to that. If a second drive were in the same

cabinet (our intended case holds two half-height drives), then it would be powered along with the primary. It would be running for the same time and be exposed to the same wear. Even archived floppy diskettes lose their data after some period of time.

My gut feeling on speed increase is an order of magnitude. That's for both access to a specific file and read/write time as well.

A hard drive plugs into the I/O expansion bus. That's the 50-pin cable header on a Model 4. The only problem that would arise is where you already have a clock board plugged into that connector. My experience has found that two peripherals plugged into that I/O bus create problems.

The operating system (LDOS/LSDOS) supports a maximum of eight drives. That can be any combination of floppies and hard drives (given supporting hardware). In the case of a standard Model 4, you can have up to four floppies and four hard drive partitions; or two floppies and six hard drive partitions.

What work doesn't demand absolute reliability? Do you run with VERIFY ON? That's one way of catching an error right when it happens. Of course, a read error can develop later caused by some later write operation. I haven't run with VERIFY ON for many years. I just don't think it necessary. And the degradation in speed is something I am unwilling to live with. I consider hard drives much more reliable than floppies. If you had to back up a full 20 megabytes of hard drive, you might spend 10-20 minutes doing it. Usually, you are just backing up modified files. That time should be no more than what you now spend in making backups - in fact it should be less.

When moving hard drives, the only caution to observe is to ensure that they are powered down and the platters have stopped rotating.

Finally, one of the reasons I decided to explore the development of a hard drive package is that I was tired of hearing all of the problems folks were having with other suppliers. And there are not that many other suppliers these days. Most of the problems folks have is with drivers. Either they are too big, or they don't work right. Well, I just got tired of trying to refer people elsewhere. We'll do the job right or we won't do it at all.

Hard disk intermittent

Fm Mike Gilien: Roy, I just finished reading my first issue of *The MISOSYS Quarterly*, and was I impressed! My only regret is that I didn't subscribe earlier, I ordered my subscription at the same time as LS-DiskDisk and am equally impressed with the program. It is a definite must for hard disk drive owners, especially those who have upgraded their hard disks to more then 5 Meg. Thanks a bunch for your continuing fine support!

I have a comment for Gary Phillips/Volume II.iv/The Hardware Corner/ Hard Disk Intermittent:

Gary, I too experienced almost the identical problems you described with my Model 4p coupled to an early R/S 5 Meg hard disk drive. I cleaned, checked, replaced until I was totally frustrated. Utilized three different drivers, and finally installed a Seagate ST-225, in case that was the problem. No joy! Finally with a lot of help and patience from a local sysop the problem was resolved. As Adam Rubin surmised, the problem lies in the Controller Board timing loop. This is almost non-existent today, but the early controllers had a means to adjust the timing loop. It's fairly simple, and straight forward to accomplish. Open the R/S 5 Meg case, and leaving the wires all attached, locate an upright capacitor in the vicinity of the center of the board. It's the only one with a small slot in/on the top of the capacitor. This is for adjustment of the timing loop. Using a small slotted screwdriver, the capacitor is adjusted in small .5 to 1 degree increments. With both units powered up, reset the computer, observe the error message (the hated 11H), make a fine adjustment (best described as light pressure on the screwdriver), then reset again. You should be using your normal boot up sequence, and this procedure works best with two people. In my case turning counter-clockwise about 1.5 to 2 degrees did the trick. Upon resetting the hard disk booted perfectly. Since we adjusted the capacitor, no more "Error 11H" messages have been encountered: and the hard disk, functions every time. There is only one other adjustable feature on the board, and that affects the stepping rate. This adjustment is on the outer edge of the board. Apparently the early Western Digital controller used in the R/S Hard Disk drives are prone to timing loop problems. The same controller board is utilized in the 5/12/15 Meg disk drives. I hope this helps in solving your problem.

XLR8er Board

Bug in RAMDISK for LDOS

Fm MISOSYS, Inc: John Gelesh forwarded a problem to me which practically caused a double take. He was trying to install the H.I.Tech XLR8er RAMDISK utility in the Model III mode. When he attempted to install it under LDOS 5.3, it always crashed during the installation. When he attempted to install it under LDOS 5.1.4, it worked perfectly. Well I couldn't believe that. So I checked it out. Sure enough, when I booted up a stock LDOS 5.3 and installed RAMDISK, it crashed at the exact point where John had stated. I booted up a LDOS 5.1.4 I had around and installed the RAMDISK; it worked perfectly. What was the difference?

I knew that LDOS 5.1.4 forced the 4Meg clock speed when it booted up; LDOS 5.3 did not. So I installed the DDT utility to get me a disassembly display with DEBUG and then hard zapped the resident 5.3 to simulate the way 5.1.4 starts up at 4Megs. Sure enough, when I installed RAMDISK, it worked.

Well this had me going for weeks until I finally stumbled onto the reason for the crash. Sometimes you can't see the forest for the trees. It was a bug in RAMDISK! When RAMDISK starts up, it first examines high memory to see if it is already installed. If so, and inactive, it re-uses the already-in-RAM image. Otherwise, it relocates an image of the driver into high memory. Seems like RAMDISK never tested for the condition of HIGH$ being at X'FFFF', i,e, nothing in high memory. It proceeded to check 65535 bytes of memory starting at address 0. Guess what?, It found a copy. But what it found was the copy just loaded! It then proceeded to use that driver at that location. That just won't work. I completely missed it in all my original testing of John's reported problem just by happenstance. When I booted up the LDOS 5.1.4, it was an old copy which had a configuration; the keyboard driver was in high memory. When I booted up the LDOS 5.3, it was a new copy without a configuration. The following patch will correct the Model III mode RAMDISK.

```
. RD51/FIX - Patch to XLR8er RAMDISK
. for Model III LDOS
. Apply via, PATCH RAMDISK/DCT RD51
X'7317'=C3 46 78
X'7846'=78 B1 CA 3E 73 21 00 00 C3 1A 73
. Eop
```

M3BOOT & MODELA/III of 6.3

Fm MISOSYS, Inc: In checking out the operation of my Revision C non-gate array Model 4 and its Model III mode operation with the XLR8er, I came across a problem with the M3BOOT utility provided with the H.I.Tech LDOS interfacing software. The M3BOOT utility loads the MODELA/III ROM image file. It essentially performs the same operation as the Boot ROM. M3BOOT uses the byte I/O DOS file SVC (@GET) but does no checking for error conditions. It then continues to fetch characters from the MODELA/III file until it finds the transfer address record. This would work if there was never an error in reading that file.

Unfortunately, the MODELA/III file which is included on the LS-DOS 6.3 master disk has an incorrect directory entry for the EOF OFFSET byte. It has a value of X'F0' but the correct value should be X'FB'. The ModelA/III file was supplied by Tandy, according to LSI. Since the COPY routine would transfer the entire last sector, the trailing end of the file was copied over to the LSI LS-DOS 6.3 master. The 4P boot ROM doesn't care about the offset since it used a full-record READ operation; it uses its own floppy I/O routines. But that does cause a problem with the M3BOOT loader which uses @GET; after the X'F0'th byte was read from the last sector, the DOS would return an EOF error, X'1C'. M3BOOT ignored this and accepted the value of 28D as a byte. It essentially locked up.

I can't give an easy patch to M3BOOT to correct this; rather, I have re-written M3BOOT to detect errors. Unfortunately, that still does not get you out of the problem; you need to correct the EOF OFFSET of the MODELA/III file in the directory. Now since the use for that is with the Model III mode, and the M3BOOT utility is for LDOS, and LDOS comes with FED, you can use FED to zap the directory. Just boot up your LDOS (without the XLR8er) and invoke FED. Specify `DIR/SYS.LSIDOS:d` as the file to edit, where ":d" specifies the drive which contains your 6.3 disk. The entry is in DIR/SYS record 10H at offset 43H. You'll find an X'F0' there; change it to an X'FB'. I have thousands of 6.3 disks already duplicated here which were provided by LSI. Therefore, it will be some time before a Level N disk gets into being. For now, those few folks who are troubled by this error, the procedure I outlined can rectify the situation for you.

5 files, 1 disk - Same name?

Fm Charles A. Ainsworth: Roy, As I told you in a phone call on the 14th., the latest is that, when running ALLWRITE, I sometimes get a new version of the same file on my floppy each time I give the Save command or that ALLWRITE does the same via the Autosave feature, which I run at a low value of 50 inasmuch as I do a lot of sensitive information which has to be recorded frequently. So after, say, a total of five saves, I end up with five files of the same name on disk. A similar problem was reported in TMQ by someone who figured that by setting the floppies at DELAY (1 sec.) the problem would be cured. That's how I am running.

Yes, I know how to overcome that by renaming all those files, removing the unwanted ones and keeping the wanted one (the one with the latest time stamp) and then renaming it back to what it should be. However, that's totally unacceptable as a modus operandi; it's far from practical, requires constant vigilance, uses a lot of disk space and I couldn't possibly expect the typist who sometimes helps me, to take care of such an arrangement. Besides, if one powers down, unaware of the multiple file situation and then later powers up to resume work on the same file, one has no idea which version one is working on. No, it's hopeless, I just can't work that way! I may be involved in a writeup on the operation of nuclear power plants or on how certain delicate surgery should be performed. Can you imagine getting a writeup like that messed up?

So, right now, my three computers are taboo for any serious word-processing, but I fear they can't stay that way for long.

I have experimented with different settings of SETI8O, but the faults are sneaky and elusive to pin down and reproduce, so I have been unable to relate things to the SETI8O parameters. However, in general, I get the impression that running SETI8O at the slowest parameters, things may be slightly better. But the lowest settings are probably giving me a slower performance than the native 4D with the 4 MHz CPU.

Another point I mentioned by phone: I played around casually with the latest HIBANKS I got from disk notes. I had also played around with the original one in the previous TMQ and disk notes. With the latter (the original version) I noted, when I invoked ALLWRITE and the logo came on screen with the filename prompt, if nothing was entered for several seconds, the logo would sort of fly to pieces and the screen would fill with garbage, (sometimes the logo would stay but the garbage would be generated opposite the filename prompt) so I put HIBANKS aside. Now, the latest revised version does something very similar so I have also discarded that. I don't get this particular problem with FIXALL and FIXBANK.

Fm MISOSYS, Inc: Charles, This confirms our telephone call of the 21st. I wanted to pass along hard copy for the three patches associated with the XLR8er speedup.

To confirm the reason for these patches, LSI changed the @CKDRV service call starting with TRSDOS 6.2 so that it became machine speed dependent. It previously had used an algorithm based on the RTC interrupt timer which was machine speed independent. Therefore, it is possible that a machine running much faster than 4MHz may have problems in recognizing that a diskette is in a floppy. There are three similar patches because the CKDRV code is present in three system files.

Incidentally, machine speed has nothing to do with control of the RS-232 hardware timing of baud rates.

```
. XLR8S12/FIX - Patch to SYS12/SYS
. Apply via, PATCH SYS12/SYS.LSIDOS
XLR8S12
D03,7C=12;F03,7C=09
D03,86=40;F03,86=20
D03,9C=40;F03,9C=20
```

```
. XLR8S2/FIX - Patch to SYS2/SYS
. Apply via, PATCH SYS2/SYS.LSIDOS XLR8S2
D00,E4=12;F00,E4=09
D00,EE=40;F00,EE=20
D01,04=40;F01,04=20
```

```
. XLR8BU/FIX - Patch to BACKUP/CMD
. Apply via, PATCH BACKUP.UTILITY XLR8BU
D17,22=12;F17,22=09
D17,28=40;F17,28=20
D17,3C=40;F17,3C=20
```

Now on the issue of the revised HIBANKS, I had it installed with ALLWRITE sitting at that filespec prompt for 24 hours (left the machine on all night and day. I didn't have any adverse behavior the whole time. So I am at a loss to understand the problem you are experiencing. Keep me posted as to the results of file access with these patches installed.

Where's the RAM?

Fm W. J. Russell: Thank you for the XLR8er chip which I ordered on the 8th April and finally arrived from Philadelphia via my friend two weeks ago. With some trepidation, I opened the lid on my model 4 26-1069A and read the manual s-l-0-w-l-y. I then followed the instructions and put the 74LS245 replacement in. (My chip had a N termination) and then assembled the board. The only problem was that I could not identify the memory chips. I am pretty sure they are not 150 nsec as I could see no appended mark but all has gone well so far. Perhaps a diagram to show where to look would be an advantage. Having read the instructions carefully I made sure the copper side of the support board was outwards. (Incidentally, as another outside USA Tandy there was no RF Shield installed around the board.) However with two boards, the screen and the circuit board there remains two ways to install even aligning the screw holes. On the third(?) try, I got it right but could not insert the bottom right corner two of the six screws as the holes were very slightly out of alignment. However it is working and the speedup is impressive. The only problem I have encountered so far is that Lescript Key Macros seem to dislike it, even when FIXALL and the filter are added. Some of the triple keys such as CLEAR/SHIFT 0 to exit have played up putting graphic characters on the screen instead of exiting.

Another small point which puzzled me at first was that the new chip seems only to have drive 0 and 1 available as the default, unlike LDOS which recognizes all three of my floppy drives without specification. Now I seem to have to do a SYSTEM (DRIVE=2,DRIVER="FLOPPY/DCT") to get the third drive. My SYSGEN and AUTO DO START are now quite busy.

I have not tried to get the hard drive going yet as this is partitioned with LDOS 6.3 and CP/M2.23. I received an LDOS/TRSDOS disk for the XLR8er but from reading the manual I think there is also a CP/M disk which should have been there. I mentioned that I also ran CP/M in the letter.

As I mentioned in the original letter, the 15 Meg Hard drive is used to run 5.3, 6.3 and Montezuma CP/M 2.23. The Hard drive is split into six drives three each for Tandy's and CP/M (with RSHard). Can I still run the LDOS 5.3 and do an install of SET180 and FIXBANKS and FIXALL as for the LS-DOS 6.3?

A different point I would like advice on. I have Toolbelt and the other day my son (15yrs) accidentally deleted a file from the hard drive in LDOS. How do you UNKILL a file on the Hard drive? There seems to be no equivalent for the floppy UNKILL. I understand the kill is only a directory flag change.

One more request, could you please send me LBMU-M4 to my home address as I am planning to develop a new journal reference database and envisage regular upkeep and pruning.

One final thing - have a good vacation. Come down to Australia some time. They say hanging upside down is good for blood supply to the brain, although I am not sure it has helped us all that much, and I have never seen a smart bat or flying fox.

Fm MISOSYS, Inc: LeScript, unfortunately, does its own keyboard scanning. It totally bypasses the DOS. That's why it doesn't quite work with PRO-WAM (although it does poll the keyboard so that the DOS spooler can be used and that allows you to activate PRO-WAM but not export back to LeScript). That's why the FIXALL filter is not effective with LeScript! Sorry, but there's nothing I can do about that.

If the XLR8er manual were to include a diagram showing you where to look on the motherboard for the RAM chips, it would probably have to have a dozen diagrams. Tandy has redesigned their board so many times, that you have to have a book full of schematics just to do troubleshooting. The socket numbers aren't even consistent. For the RAM, though, its probably just as easy to look for eight identical chips in a row which also have a second row (filled if your machine is 128K).

In the case of two drives versus four drives, you must be referring to LS-DOS, not the "chip". For more years than I have fingers on a hand (and I have a full set), TRSDOS 6 defaults with drives 2 and 3 disabled. Actually, you have a simpler command to enable drive 2;

```
SYSTEM (DRIVE=2,ENABLE)
```

The boot-up drive code table has the proper information in it for a standard drive 2 floppy (physical third). Just enable it then SYSGEN. You could also patch BOOT/SYS to achieve the same thing. That information was publicized some time ago. In fact, it was published on page 61 of *THE MISOSYS QUARTERLY* Volume I, Issue iii (Winter 1987).

The XLR8er comes with interfacing software for one DOS; we have software available for DOS 6, LDOS, and CP/M. But you have to order each additional for $9.95 each.

As far as an UNKILL for hard drives, there's no difference in what needs to be done over floppies. The UNKILL which is part of our UTILITY DISK I will do the trick. Actually, there is more done that just resetting one bit when you KILL a file. The active bit is reset in its directory record, the directory entry code (hash code) is removed from the hash index table, and the space allocated to the file is deallocated from the granule allocation table. An UNKILL facility must revers the procedure after ensuring that all granules previously used by the file are still unallocated.

We're also saving our money. One day we will get down under. Until then, G'day (is that how you would write it?).

International checksums

Fm Bob Hyde: Thank you for TMQ II.iv. It arrived on the last day of May. I am interested in the discussion on the XLR8er board and its compatibility. I have a Model 4P international keyboard. It has a model number 26-1080A, serial no.004311. It has 128K and a Tandy hires board and I hope to keep it going with the aid of you Sterling folk. I do not know what is inside, gate/non-gate etc., but when I run MEMTEST/CMD (TRSDOS 1.3i with MODELF/III) it suggests that all is not well. To be precise: ROM check A should be 9639. It is 9034. ROM check B should be 407C. It is 3FA3. ROM check C should be 3BE7 or 10A6. It is 4022. Alternate character test is OK. Query. Is this all bad? Please don't suggest asking TANDY, they hadn't heard of TRSDOS last week. They recently changed staff.

Fm MISOSYS, Inc: It sounds like you have a gate array 4P; it's the "A" suffix which gives me that clue. In any event, the XLR8er should work with any version of the 4P - even with the hires graphics. The XLR8er mounts in the modem slot; thus, you can't use an internal modem in that arrangement.

As far as the "ROM" checksums, I doubt that you have cause for alarm. For one thing, there is no BASIC ROM in your computer, only a boot ROM. The "simulated" BASIC ROM is actually the MODEL?/III image file you load during the Model III mode startup procedure. Since you have an international version of the MODEL?/III file, I would really doubt that the checksums would correlate with the numbers you are comparing against. The TRSDOS 1.3 MEMTEST probably uses checksum values that match up with a standard ROM from a Model 4.

XLR8er & Speedup

Fm John Cerul: Roy, Pro-Enhcomp has been received, and I am checking it out. Please ship me a copy of your MS-DOS catalog. No I haven't jumped on the bandwagon, one of my sons lost a wrestling match with a lawn-mower and lost parts of two toes so he has decided to go back to school. He wants to study programming, so I bought him a 1000SX to study with.

That raised cain with my budget so I'm glad that you retained sales of the XLR8er, I want another one, but I can't bend the budget any more, for at least a couple of months. I am holding back for that harddrive though.

I have installed the one XLR8er that I have in my new? $599.00 4D and it has become my favorite. Its what I'm using to check out Enhcomp. Listed below are various timings of Larry Hildebrand's sieve (TMQ II.iv) run on the 4D with the XLR8er under Enhcomp.

No switches	807sec.
-NX	753sec.
-NS-NX	726sec.

I have also installed the Smartwatch Mods in all 4 of my machines.

The fix for my previously reported problems with the scroll stop (left <shift><@>) is very simple. I just had to delete (wait=1) from the line;

```
set *ws fixall/flt (wait=1)
```

in the HI-TECH SETUP/JCL file.

I was also having problems with the PLOT statement of Enhcomp. It would not draw a horizontal or diagonal line, and wasn't drawing the proper length vertical lines. I had HIBANKS installed. When I dropped back to the "stock" setup with the RAMDISK driver, PLOT worked OK. So I'll wait till receipt of Disknotes II.iv to reinstall HIBANKS.

I'm enclosing an item from "Northern Bytes" that I found useful on my two old style Mod 4's. I put the mod in both and haven't been "bitten" yet. The old style 4's are now running at 4 MHZ as close as I can tell.

Thanks for the fine service up to now. Hope you can continue to hang in.

Fm MISOSYS, Inc: Yes, the II.iii issue version of HIBANKS would effect the EnhComp PLOT statement since it uses the alternate registers for speed. That HIBANKS also used the alternate registers during interrupt processing. The II.iv version should fix that up.

Thanks for the article reprint. See the following topic for that.

Model 4 Speedup

Fm MISOSYS, Inc: Coincidentally, three different people recently sent me copies of the following article. Perhaps because some of the TRS-80 computer clubs are beginning to reprint it. In any event, because of this exceptional circumstance, I thought that TMQ readers should also be aware of it.

Fm Dave Krebs: Roy, The following is a little info on some of the 'early' Model 4's (I have one of 'em) relating to their true CPU speed. You probably are aware of it, but I thought it may be of some interest to some of the TMQ readers ... Pitch it if you think it isn't suitable for TMQ.

Quite some time ago (read years), an article appeared in the 'Micro Info Exchange', the newsletter of the Cabrillo Computer Society. It was written by Dave Owen, CSS Tech Specialist.

Since my Model 4 had a date on the case of FEB-83, I was interested in the subject of speeding up my CPU. I had noticed that mine didn't seem as fast a friend's 4P. I removed the cover to check out the part number of U3, and not finding a matching number, buttoned it up again. That was a couple of years ago.

Recently, I became aware of the software and instructions to add the SmartWatch to a Model 4, so I decided to take another look at PAL U3. This time I ran a little BASIC for-next loop test before and after I bent up pin 7 of U3. The following is the routine:

```
DEFINT A-Z : CLS
PRINT@(10,35),TIME$;
FOR I = 1 TO n
 FOR J = 1 TO 10000
 NEXT J
NEXT I
PRINT@(13,35),TIME$
END
```

I ran this routine with a value of 5, 10 & 20 for 'n', the results were as follows:

n=	Before	After	Diff	Improve
5	0:26	0:21	0:04	15%
10	0:51	0:41	0:10	20%
20	1:43	1:22	0:21	20%

So far, I have only found two small problems: (1) I have some software with timing and/or blinking routines and now they're

too fast, I had to increase the loop counters and (2) A couple of applications have sound (beeps) and the durations were shortened and pitch made higher and I have trouble hearing them. The BASIC statement, SOUND 0,4 is now SOUND 0,2 for about the same beep.

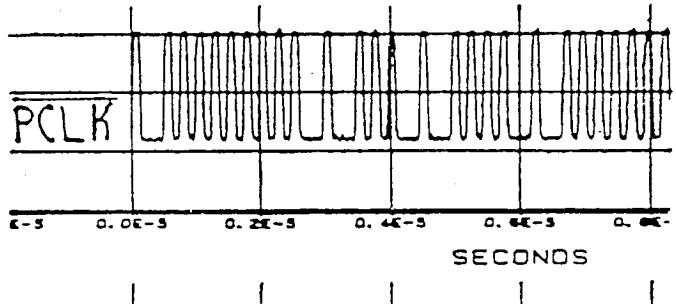
The following is the original article.

FAST FIX FOR MODEL 4 SLOWS
By Dave Owen, CCS Tech Specialist

[Reprinted from the Micro Info Exchange, newsletter of the Cabrillo Computer Society].

The Radio Shack Model 4 Computer is advertised to operate at a two- or four- megahertz clock rate. Our investigation has revealed the truth of this claim. the Tandy specifications do not, however, state the fact that the master clock does not always "tick". Because of this persistent miss, the machine operates at approximately 3.37 MHz ... or nearly one-sixth slower than claimed.

The pre-programmed PAL (programmable array logic in the back panel's U3 socket), U3 logically ties the op-code fetch signal (M1) to the crystal oscillator output (*PCLK). The result is that clock cycles are suppressed during each op-code fetch on those PALs labeled with Radio Shack part number 8075166. A later version PAL has reportedly eliminated this design feature. But most of those sold in 1983 and earlier do run below specs.



This shows component M1 causing PCLK in U3 to also output erratically, so that the missing pulses mean, on balance, slower speed.

The clock suppressions aren't needed with the Z80A (the computer was originally designed to operate with a ZILOG Z800 that was never released). If you have this particular PAL, the solution is simple: Remove IC chip U3 by prying it up at each end with a small screwdriver. With a pair of small pliers, bend pin 7 horizontal so it will no longer sit in the socket. Replug this chip back into your board. Presto, a 4 MHz operating speed.

Does this really make a difference? Software, particularly the operating system, is "tuned" to certain operating speed environments. When formatting my 8-inch drives under TRSDOS 6.2, for example, the time required dropped from 8 minutes to just over 2 minutes. Other operations also work more efficiently, such as mathematical calculations, We times FOR-NEXT loops at 16% faster, try it, you'll like it, and it's free!

Hardware Tips

Fm John P. Eich: To *THE MISOSYS QUARTERLY* Editor, I submit the following tips:

#1. While installing an XLR8er board in my Model 4P, I found that the board falls out of the modem slots. I remedy this by stripping the insulation off of a piece of teflon insulated stranded wire (0.080" OD., 16 gauge I believe). I used 2 pieces, each 3.5" long, of the insulation. Tucking a piece into each of the two plastic guides and centering each in the modem slots. The XLR8er fits fine now, won't fall out and the added strips stay firmly in place.

#2. While working up a program or reviewing data, I've often wished I could look back at a screen full of information. I've found a way! In the 4P gate array system, U82 is a 6116 static RAM which contains the information presented on the screen. Pin 18 on that chip is the CE (chip enable) signal and is wired to ground such as to keep the chip always working. I piggy backed another 6116 soldering all pins together except pin 18 (lifting pin 18 of the original out of its socket). I then mounted a SPDT (a tiny one) in the case just below the reset switch, there is room enough but you must measure carefully after removing the computer face. Two wires run to the two pins 18 and a third from the throw connection of the switch to ground on the computer board. I presume you understand the two wires came from the outer poles of the switch. It works fine! If you want the "second" screen clear though, on boot up (assuming a warm system) as soon as the system clears the screen, throw the switch and press the reset switch again - now you can start working. When you have a few lines or a screen full that you'd like to refer to later, throw the switch and keep on. Remember thought that this does not reset the line or character count so you'll be starting at the spot you left the first screen but it will scroll, etc., normally. I should mention that each of two pins 18 will need a pull up resistor to +5V. One tenth or quarter watter will do fine to a +5V trace nearby.

The switch I used has a center position (off). I thought (without thinking of course - it was the only switch I had handy) this would be fine - I could use this position to blank

the screen when I walk off and let the machine run - not so! The screen fills with junk! Investigating the board, I found that I could use either of two empty (not used) gates of U92, a 74LS00 NAND gate IC (2 highs = a low). So, I've run wires from each of the pins 18 of the two 6116s (remember they will be pulled up if the switch is off) to pins 4 & 5 of U92. Pin 6 will go to a third 6116 pin 18 (it won't need a pull up resistor) and this should give me three screens with two to save. I haven't mounted that third 6116 because I first must check clearance. I may have to eliminate the socket.

#3. I'm a fanatic when it comes to working on circuit boards. I run my wires on the Socket Side of the boards. When connecting to sockets or IC pins, I desolder (using small wire wick) the pin and straighten it (Radio Shack bends them to hold things in when wave soldering). When attaching a wire to a pin, I make a very small loop by taking a turn around a small needle, cutting off more than one turn and looping the wire around the pin. I hold the wire in place with a small lump of auto body putty. It stays soft for ever. I've used the same lumps for years. It's plastic and sticky enough that when pressed over a wire, it holds it in place adequately. Of course I remove it after it's served its purpose.

I'm fortunate enough that I still have some of the very very small hollow rivets we use to repair circuit boards. When you need to "surface" (bring a connection to the top of a board) a wire, I use one of these units. The ones I use have an O.D. of 0.060" and smaller; with a flange on one end and should be only long enough to come through the board. Hold your board to a bright light and you can find places you can use a rivet and miss the traces on both sides. Drill a hole just large enough to hold the rivet in place. Take a larger drill and with your fingers, twist it to break the sharp edge of the hole. On the side you want to expand the rivet, make a small depression with the drill. Insert the rivet from the other side, back it up with the flat end of a large drill bit (or rod), held in a vice and tap it with a center punch. It's now permanent and neat. Run your wire (I use the so-called wire-wrap wire) to the hole, strip the insulation, pull it tight and bend it to hold it in place.

Now for a quick disconnect. I use the pins from damaged (the good pins that are left) "machined pin" or "high reliability" sockets. Each of these pins have an excellent male end, especially the gold plated fingers in the female end) and a female end. You can stick the male end in the rivet I mentioned above and solder it in. Don't let any solder get in the female end. Now strip the insulation from the end of a small stranded wire, about 1/16" will do. Insert it in the female end of another socket pin and solder it. Slip a piece of shrink tubing (about 0.10" O.D.) over the wire and all but the male end of the pin. Shrink it and you now have a subminiature disconnect.

I think some folks can use this help.

Floppy Drives

Mixing 1 & 2 sided

Fm Gary Shanafelt: It goes without saying that I'm flattered to see my article on converting TRS-80 games into CMD format printed out so nicely in the last issue of the Quarterly. If there are any comments from readers, I will forward them on to you if they mail them to me direct; if they post them on CompuServe, I suppose I will read them in the next issue.

I would like to add a word on my experiences adding a double sided drive to my Model 4. What Joe says to John Gelesh on page 82 is true as far as it goes; but it applies only to adding TWO drives. I recently added just one, drive 1, when a friend replaced his original drive in a PC-XT with a 3 1/2" unit and gave me the old drive. I bought a new cable with all the pins in it from Aerocomp, and pulled the pins that affect the drive selection (pin 12 on the drive 0 connector, pin 10 on the drive 1 connector, pin 14 on both connectors). My original RS drives have no jumpers on them; they are interchangeable according to cable position. The PC-XT drive was apparently already set to be drive 1; it was drive B in my friend's computer.

Anyway, once everything was hooked up, the PC-XT drive would still only format as a single sided drive -- even though pin 32 was present on the cable. A call to Aerocomp tech service solved the problem -- pin 32 is apparently used by the RS drives as part of the drive select process, so it must be pulled from drive 0 if I want to keep that drive standard RS but get DD use from drive 1 where pin 32 is NOT pulled. The resulting cable configuration, which works just as I want, is with pins 12, 14, and 32 pulled for drive 0 (standard RS single sided drive), and pins 10 and 14 for drive 1 (the double sided PC-XT drive). Of course, if I ever replace drive 0 with a double sided unit, I will have to replace that pin 32 to get full use from it.

I want to thank Aerocomp for their help in getting this system up and running, and if I do buy a new drive 0 it is obvious where I will go to get it!

3.5" Drive	No plans to write driver for Model 4	I. iv 44
80 MICRO	General Comments	I.iii 23
80 MICRO	Letter by Gary Phillips	I. iv 18
@CMNDR	A new DOS service call	I. ii 14
@EXMEM	Extended Memory Management	I.iii 75
@EXMEM	New @EXMEM SVC, etc.	I. iv 17
ADE	Difference to diskDISK	I. ii 60
ADE	Product Highlight	I. i 64
AlphaTech	More on AT patch	I.iii 109
AlphaTech	Notes on the AlphaTech memory board	I. i 86
AlphaTech	Problems with DOS x.3	I.iii 28
AlphaTech	RAMDRV/LQR - Driver for AlphaTech boards	I. i 87
AlphaTech	SET2RAM and MEMDISK driver	I. iv 21
AlphaTech	SYSATFIX and AlphaTech board	I.iii 108
ANSI	ANSI Standards - Where to get them	I. iv 15
ANSI	What is it	I. iv 47
ASSEMBLER	Q & A	I.iii 85
ASSEMBLER	RELOCATABLE ASSEMBLERS by Roy Soltoff	I. i 52
ASSEMBLY	8086 Assembly Language for Z80 programmers	I. iv 71
AUTO	How to AUTO with hard drive	I. ii 28
BANKER	Line 00052 of listing in I.ii p.34 is incorrect	I.iii 27
BANKER	The BANKER by Roy Soltoff	I. ii 33
BASIC	!<ENTER> as equivalent to SYSTEM,ENTER>	I.iii 58
BASIC	BASICSVC by Robert M. Connors	I. i 50
BASIC	Cursor strangeness	I. iv 40
BASIC	Data Indexing - by Paul Wade, Vic. AUS	I. iv 62
BASIC	Enhancements	I. ii 17
BASIC	EnhComp brief description of some features	I. i 22
BASIC	Length of array names used to pass param. to SVC	I. iv 44
BASIC	MORSE Code from a Model 4	I. iv 60
BASIC	New version of BREF which handles Line Number 0	I.iii 60
BASIC	STRIP/BAS - by James Beard, Ph.D.	I. iv 59
BASIC	UNLOCK by Peter Lengsfeld	I. i 45
BINCALC	Programmers Calculator by Bryan W. Headley	I. i 81
BREF	New version which can handle line number 0	I.iii 60
BSORT	Problems with 2-dimensional array	I. ii 61
BSORT	Product Highlight	I. i 64
C	GRPLIB	I.iii 22
C	LC superceded by MC	I. ii 68
C	SPLITLIB/CCC by Rich Deglin	I. ii 39
C	The C Language. MC announced (Version 1.5)	I. i 25
CAT	Provision of the CAT command under LDOS 5.3	I.iii 67
CATALOG	Policy of supply	I. ii 5
CATALOG	Price List effective April 1, 1987 and July 1, 87	I.iii 9
CIS	Requests for CIS files on disk	I. iv 15
COMPUSERVE	Forum	I.iii 30
COMPUSERVE	FORUM commands	I. iv 103
COMPUSERVE	Forum policy on deleting messages	I. iv 103
COMPUSERVE	Garbage messages	I. iv 103
COMPUSERVE	Problems accessing a BBS	I. iv 103
COMPUSERVE	Requests for CIS files on disk	I. iv 15
COMPUSERVE	SIG paging	I. iv 103
COMPUSERVE	Unsqueezing a file	I. iv 104
CONVCPM	Product Highlight	I. i 65

CORRESPONDENCE	How it's dealt with	I. ii 9
DD&T	Problem when LS-diskDISK is installed as well	I. i 65
DED86	a disk/file/memory editor	I. ii 18
DED86	MS-DOS based disk editor	I. iv 78
DESCRIBE	Product Highlights	I. i 65
DIR	DIRectory command defaults	I. i 16
DISK NOTES	Cannot be ordered for shipping together with TMQ	I.iii 14
DISK NOTES	Contents of DISK NOTES 8	I. iv 4
DISK NOTES	What they are all about	I.iii 3
diskDISK	Difference to ADE	I. ii 60
DOS	Am I at DOS Ready	I. i 15
DOS	Notes from our Compuserve SIG	I. i 16
DOSPLUS	Incompatibility Note	I. i 13
DOUBLE PRECIS'N	The case of ACCURACY vs PRECISION	I. ii 28
Drive	2-sided drives	I.iii 105
Drive	FDC 1793 disk controller from Western Digital Info	I.iii 108
Drive	TEAC FD-55BV-06 drives for IBM PC	I.iii 105
DSM	Disk Sort and Merge	I. iv 78
DSM	Product Highlight	I. i 65
ED/ASM-86	80x86 assembler	I. iv 79
ED/ASM-86	MS-DOS assembler	I. ii 20
EDAS	Assembly Language: EDAS & MRAS	I. i 21
EDAS	EDAS and PRO-CREATE	I.iii 93
EnhComp	BASIC Line #'s in Z80 MODE	I.iii 95
EnhComp	Bits & Pieces	I.iii 93
EnhComp	Brief description of BASIC compiler	I. i 22
EnhComp	Data Table length values issued during a compile	I. iv 83
EnhComp	Disabling RMARGIN in EnhComp	I.iii 95
EnhComp	Incompatibility with MS BASIC interpreter	I. iv 82
EnhComp	not set up to use PR/FLT	I. iv 80
EnhComp	Patch to solve problem with PRINT USING	I. iv 83
EnhComp	PRINT: with EnhComp	I.iii 94
EnhComp	Problem with "If expression THEN"	I. iv 81
EnhComp	Problem with SOUND	I. iv 80
EnhComp	Program devel. almost as easy as interpretive BAS	I.iii 16
EnhComp	Q & A	I. ii 61
EnhComp	USING	I. iv 82
Family News	Comments	I. iv 17
FastBack	When to use it	I.iii 95
FDC 1793	FDC 1793 disk controller from Western Digital Info	I.iii 108
FED II	Product Highlight	I. i 66
FILTER	Writing FILTERs for LSDOS 6.x by Roy Soltoff	I. iv 64
FM	FM - File Manager	I.iii 95
FORMAT	Add 2-side prompt to TRSDOS 6.2.x FORMAT	I. i 16
FORMAT	Bug with 2-sided drives LDOS 5.1 and TRSDOS 6.2	I. i 15
FORMS	Q & A	I. ii 26
FORTH	Licensing rights	I. i 39
FORTRAN	Bug in Mod. 4 with file LRL	I.iii 15
FORTRAN	Enhancements by Harry G. Clayton, Jr.	I. ii 44
FORTRAN	Observations re ACCURACY	I.iii 15
GOBBLING BOX	A game for I,III and Mod.4	I. ii 11
GW BASIC	Clear to end of frame	I. iv 47
Hard Drive	Hard drive problems	I. iv 42
Hard Drive	Harddisk Crash	I. i 16

Hard Drive	HD Contrpllers	I. ii 85
HD64180	HD64180 Information	I.iii 108
HYPERSOFT	Model 4 Emulator for PC-compatible machines	I. iv 44
IFC		I.iii 95
JCL	Control Characters acceptable through @KEYIN	I. ii 79
JCL	LCOMM problem	I.iii 33
JCL	Problems with Ending JCL files	I.iii 56
KILL	Provision of KILL command under LSDOS 6.3	I.iii 67
KSMPlus	SYSGEN KSMPlus?	I.iii 95
LB	Conversion of PowerMAIL to LB	I. i 68
LB	Converting MailFile to Little Brother	I.iii 69
LB	FORMS Filter, Conversion of BASIC files	I. i 66
LB	LB and LSDOS 6.3	I.iii 96
LB	LB and MS-DOS	I.iii 96
LB	Question re backup	I. ii 4
LB	Upgrading Little Brother to LSDOS 6.3	I.iii 68
LCOMM	Problem when invoked by JCL	I.iii 33
LDOS	Backup problems with doublesided drives	I. ii 25
LDOS	Caution for Directory Tinkerers	I. i 14
LDOS	Obscure overlay problem	I. i 18
LDOS	SYSTEM (FAST) and the 4P MODELA/III	I.iii 41
LDOS	300 rpm, keyboards and other matters	I. iv 26
LDOS	KSM bug - for years!	I. iv 25
LDOS	LDOS Literary Section	I. iv 19
LDOS	LDOS on Model 4 and cursor speed	I. iv 24
LDOS	SET2RAM and MEMDISK driver	I. iv 21
LDOS / LSDOS	What's the version	I. iv 21
LDOS 5.1.4	Problem with single drive Model I	I. ii 24
LDOS 5.1.4	SYSGEN the CLOCK	I. i 13
LDOS 5.1.4	Update Documentation	I.iii 40
LDOS 5.3	80 MICRO Review of LDOS 5.3	I. iv 20
LDOS 5.3	@DODIR	I.iii 48
LDOS 5.3	@PRT documentation error	I.iii 47
LDOS 5.3	Abbreviated README/TXT re renaming LBASIC to BASIC	I.iii 45
LDOS 5.3	BSORT51 and LDOS 5.3	I.iii 93
LDOS 5.3	Creating a NULL-length file	I. iv 31
LDOS 5.3	Date-not-current flag	I.iii 37
LDOS 5.3	Defaulting FORMAT to SIDES=2	I.iii 51
LDOS 5.3	DIR (A=N) patch	I.iii 38
LDOS 5.3	DOCONFIG/CMD	I.iii 36
LDOS 5.3	Double sided boot disk	I.iii 39
LDOS 5.3	DUAL - and other LINKs	I.iii 47
LDOS 5.3	Expectations	I. ii 6
LDOS 5.3	FEDing the DIR/SYS file	I.iii 48
LDOS 5.3	Future	I.iii 40
LDOS 5.3	Installing LDOS 5.3 on hard drives	I. iv 28
LDOS 5.3	KILL Command	I.iii 35
LDOS 5.3	MAX-80 ROM image	I.iii 38
LDOS 5.3	Miscellaneous Problems	I.iii 50
LDOS 5.3	MISOSYS sole authorised vendor of upgrade kit	I.iii 3
LDOS 5.3	Model 4 Hardware Interface Kit	I.iii 6
LDOS 5.3	Model III Cursor and CTL key in LDOS	I.iii 42
LDOS 5.3	NEWCLOCK and x.3	I.iii 39
LDOS 5.3	Old Passwords	I. iv 20

LDOS 5.3	PRFLT/FIX error and SET2RAM	I.iii 46
LDOS 5.3	Printer Control and PR/FLT	I.iii 35
LDOS 5.3	Problem with TED/CMD	I.iii 42
LDOS 5.3	README/TXT corrections	I.iii 38
LDOS 5.3	Reset	I.iii 35
LDOS 5.3	SLOW BACKUP and Interleave	I.iii 35
LDOS 5.3	SMOOTH What's it for	I.iii 36
LDOS 5.3	SUPERLOG and LDOS 5.3	I. iv 25
LDOS 5.3	SYNONYM adapted for LDOS 5.3	I. iv 24
LDOS 5.3	SYSGEMing other configs	I.iii 46
LDOS 5.3	SYSGEN of SMOOTH	I.iii 36
LDOS 5.3	SYSTEM (SYSTEM=d) during JCL	I.iii 39
LDOS 5.3	TED problems?	I. iv 30
LDOS 5.3	The JCL //EXIT gotcha	I. iv 30
LDOS 5.3	The TIME? prompt	I.iii 37
LDOS 5.3	Timekeeping with 50Hz	I.iii 49
LDOS 5.3	Trouble with DEVICE	I. iv 23
LDOS 5.3	UN-DATECONV	I.iii 48
LDOS 5.3	UNDATE - by Luis M. Garcia-Borro	I. iv 56
LDOS 5.3	Use of CMD"V ..." in BASIC	I.iii 36
LDOS 5.3	What won't work on x.3	I.iii 38
LDOS 5.3	Where"d the CAT go?	I.iii 37
LDOS 5.3	Where'd @ABORT go?	I.iii 49
LDOS 5.3	Where's MODELA/III?	I.iii 37
LDOS 5.3	Where's the missing file?	I.iii 48
LDOS 5.3.0	Upgrade Kit	I. ii 13
LDOS 5.x	Caution about fixes in I.i p.14	I. ii 8
LDOS 5.x	SVCs and whatnot	I. i 14
LDOS SIG	Q & A	I. ii 29
LEAP YEAR	Is 2000 a Leapyear?	I.iii 24
LEAP YEAR	Leap years a la 2000 revisited	I. iv 13
LED	Editors: LED	I. i 39
LED	Problem with hanging up when encountering 00 byte	I. ii 67
LIB	LIBrary command changes	I. ii 14
LISP	No implementation for TRSDOS 6	I.iii 16
LISP	Poem by Guy L. Steele, Jr.	I.iii 28
LSDOS 6.2	Harddrive problem	I.iii 52
LSDOS 6.2	LS-DOS 6.2 for the Model II/12	I. i 12
LSDOS 6.2.x	*PR timeout	I.iii 52
LSDOS 6.3	@DODIR	I.iii 48
LSDOS 6.3	BASIC & @SOUND	I. iv 32
LSDOS 6.3	Bug in TED/CMD	I.iii 56
LSDOS 6.3	Compliments	I. iv 13
LSDOS 6.3	Correction of error in patch published in I.iii.65	I. iv 45
LSDOS 6.3	Disk driver function codes	I. iv 33
LSDOS 6.3	Fix for TAPE100	I. iv 32
LSDOS 6.3	NEWCLOCK	I.iii 57
LSDOS 6.3	One-character device names	I. iv 40
LSDOS 6.3	Patches	I.iii 65
LSDOS 6.3	Patches to upgrade to LSDOS 6.3.0 Level "I" or "J"	I. iv 45
LSDOS 6.3	Program revisions for LSDOS 6.3	I.iii 56
LSDOS 6.3	Q & A	I.iii 61
LSDOS 6.3	Q & A	I.iii 56
LSDOS 6.3	RamDrive driver & LSDOS 6.3	I.iii 110

LSDOS 6.3	Special Characters	I. iv 32
LSDOS 6.x	BASIC's EOF( ) Problem	I.iii 53
LSDOS 6.x	FORMS questions	I. iv 37
LSDOS 6.x	IBM PC diskettes	I. iv 34
LSDOS 6.x	Keyboard Filters	I. iv 42
LSDOS 6.x	Writing FILTERS for LSDOS 6.x by Roy Soltoff	I. iv 64
LSI	More on the aquisition of LSI	I. i 2
LSI	Mutual funds	I. iv 45
LSI	Policy on revisions	I.iii 58
LSI Column		I. iv 44
LSI Column		I.iii 57
MACH2	Suggested utility to place files in desired sector	I. ii 27
MACHINE SENSING	by Jeffrey R. Brenton	I. i 55
MACRO	Macros by Timothy Adye	I. i 44
MARK III COLL.	2-disk collection of documented software	I.iii 8
MARK IV COLL.	2-disk collection of documented software	I.iii 8
MAX-80	MAXIMUL is the user group for MAX-80 users	I. ii 3
MAX-80	Moving a Hard Drive's directory	I. i 13
MAXIMUL	Usergroup for MAX-80 users	I. ii 3
MC	A CC for MC by Mike Bedore	I. i 30
MC	Accross the C	I. ii 23
MC	C compiler changes for x.3	I. iv 83
MC	CC/CCC on Disk Notes 6	I. ii 4
MC	Clarification of difference to LC	I.iii 88
MC	Compiling separate fields	I.iii 88
MC	Getchar( ) and the ROM keyboard driver	I.iii 90
MC	Initialisation of automatic variables	I.iii 90
MC	MC and #include variations	I. iv 92
MC	MC and bug reports/fixes	I. iv 92
MC	MC and C variables	I. iv 89
MC	MC and GREP speed	I. iv 83
MC	MC and separate compilation	I. iv 84
MC	New full-C compiler released 07.10.86 (Vers. 1.5a)	I. i 25
MC	On assigning character pointers	I.iii 88
MC	Patch to fix Wildcard problem	I.iii 87
MC	Pending MC bugs	I.iii 92
MC	PRO-MC and EOFs	I. iv 89
MC	PRO-MC and variable initialisation	I. iv 85
MC	PRO-MC and __LINE__	I. iv 88
MC	Problem with WINDOW/CCC (part of PRO-WAM)	I. iv 96
MC	Problems with MCOPT	I.iii 91
MC	Q & A	I. ii 68
MC	Retrieving Date/Time with MC	I.iii 87
MC	SIDEWAYS with MC	I. iv 88
MC	Speeding up MCP Preprocessor	I.iii 91
MC	SPLITLIB/CCC by Rich Deglin	I. ii 39
MC	Structures and bit fields	I. iv 91
MC	UNREL : REL to ASM translator	I. ii 11
MC	Update on MC libraries	I.iii 92
MC	Using I/O redirection with MC for debugging purp.	I. iv 64
MEMDISK	SET2RAM and MEMDISK driver	I. iv 21
MISOSYS	Contributions to TMQ	I. i 5
MISOSYS	Letter of appreciation from France by Michel Houde	I. iv 19
MISOSYS	Software Submissions	I. i 5

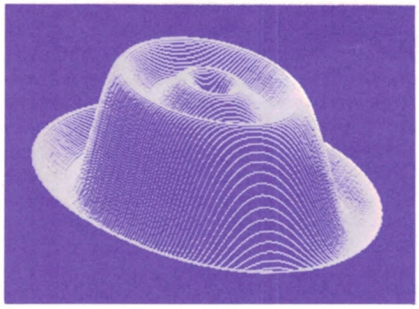
MISOSYS	Upgrading of MISOSYS Products	I. iv 15
MISOSYS	Where's MISOSYS been?	I. i 4
Model 4	BOOT a Model 4 from a HARD DISK	I.iii 107
Model 4	Gate vs Non-Gate	I.iii 105
Model 4	Harddisk boot problem solved with HDBOOT/DOC	I. ii 27
Model 4	Help with RS-232 specs	I.iii 105
Model 4	Inverse Video	I. i 11
Model 4	Megadisk driver in High Mem - Misbehaved software	I. ii 28
Model 4	Model 4 RAM upgrade - Case I	I.iii 107
Model 4	Model 4 RAM upgrade - Case II	I.iii 107
Model 4	Port control under LDOS	I. i 11
Model 4	Stacking Commands	I.iii 52
Model 4	Top Of Form feeds	I.iii 52
Model 4/4P/4D	@EXMEM - Extended MEMORY Management	I.iii 75
Model 4P	Double sided drives	I. ii 85
Model 4P	Keybounce	I. ii 86
Model 4P	The 4P BOOT ROM by Roy Soltoff	I. ii 87
Model I	CPU Problems	I.iii 106
Model I	re LDOS 5.3 beyond Dec.12, 1987	I.iii 17
Model II	LS-DOS 6.2	I.iii 53
Model III	2-sided drives	I.iii 105
Model III	Bored of boards?	I. ii 87
Model III	Keyboard problem with new C-ROMs	I. ii 27
Model III	Model III Network ROMC	I.iii 111
Modem	NULL Modem - The great Communicator	I. ii 23
MORSE	BASIC program for Morse Code practicing	I. iv 60
Mr ED	Backing up Mister ED	I.iii 102
Mr ED	Q & A about VED/APP	I. ii 80
MRAS	Assembly Language: EDAS & MRAS	I. i 21
MRAS	Mixed Case + PRO-MRAS	I.iii 96
MRAS	Mixed case + PRO-MRAS	I.iii 86
MRAS	MRAS switches -CI and -GC	I.iii 86
MRAS	MRS510 missing in TMQ I.ii - will be in I.iii, DN7	I.iii 96
MRAS	obscure bugs to be corrected by later re-assembly	I. iv 95
MRAS	Q & A	I. ii 73
MRAS	Q & A	I. iv 94
MS-DOS	Getting into MS-DOS Assembly	I.iii 84
MS-DOS	HEX Listings for MS-DOS	I.iii 31
MS-DOS	SIG	I.iii 32
MS-DOS	ANSI - What is it?	I. iv 47
MS-DOS	Editor kudos	I. iv 48
MS-DOS	HELP with EXE files	I. iv 49
MS-DOS	LDOS/MS-DOS Utility	I. iv 48
MS-DOS	PC-DOS vs MS-DOS	I. iv 48
ORDERING	How to go about it via Compuserve	I.iii 31
PaDS	Partitioned Data Set Utility	I. iv 95
PaDS	Product Highlights	I. ii 78
PaDS	Product Highlights	I. i 70
PaDS	Q & A	I.iii 97
PARMDIR	Product Highlight	I. i 75
Patch	Means of getting patches	I.iii 26
Patch	Procedure how to install	I.iii 58
PIRACY		I.iii 22
PIRACY	Analysis of the problem	I. ii 2

PIRACY	Responses to TMQ I.ii Blurb	I.iii	12
POKEDO	Adaptation of POKEPR	I. ii	3
POKEPR	by Roy Soltoff - Poke values to printer f. DOS Rdy	I. i	45
POKEPR	Correction of mistake in original listing	I. ii	3
PRCTRL/APP	PRO-WAM Appl. customisable to ANY printer	I. ii	81
PRICE LIST	effective April 1, 1987 and July 1, 1987	I.iii	9
PRO-CREATE	$-$	I. iv	78
PRO-CREATE	Close fault error	I.iii	93
PRO-ESP	ALTRES	I. iv	78
PRO-ESP	PRO-ESP's CVT324 and listing	I.iii	97
PRO-ESP	Product Highlight	I. i	75
PRO-WAM	A new name for PRO-NTO	I. i	77
PRO-WAM	Can it be invoked by JCL? - Use PRUN/CMD instead	I. ii	79
PRO-WAM	PDS	I. ii	80
PRO-WAM	PRCTRL application - can be cust. to suit any pr.	I. ii	81
PRO-WAM	PRO-WAM and "bigram"	I.iii	103
PRO-WAM	PRO-WAM and AUTODIALER	I.iii	104
PRO-WAM	PRO-WAM and PSORT	I.iii	100
PRO-WAM	PRO-WAM Release 2 description of features	I. iv	9
PRO-WAM	PRO-WAM's BRINGUP/APP past 1987	I.iii	103
PRO-WAM	PRO-WAM/PRO-NTO - What is it	I.iii	101
PRO-WAM	Problems with LeScript	I.iii	104
PRO-WAM	Removing PRO-WAM from Memory	I.iii	102
PRO-WAM	Reusing the screen with PRO-WAM	I.iii	103
PRO-WAM	Trade In offer for 'old' PW users (exp. 03/31/88)	I. iv	12
PRO-WAM	Use with Alpha Memory Board	I. ii	79
PRO-WAM	WINCALC/APP by Bryan W. Headley	I. i	81
PRO-WAM	PRCTRL/APP: Appl. customisable to ANY printer	I. ii	81
PRO-WAM	Problem with KSMPlus	I. iv	96
PRO-WAM	PSORT in original version	I. iv	96
PRO-WAM	single key 'hot key'	I. iv	96
Profile 4+	TYPEIN & Profile 4+	I.iii	98
Public Domain	RAMDRIVE - PD driver on COMPUSERVE	I.iii	29
RADIO SHACK	Problems of getting Pro-WAM through Tandy/RS store	I. ii	81
RAMDRIVE	Public Domain driver on Compuserve	I.iii	29
RAMDRV/LQR	Driver for AlphaTech boards	I. i	87
RATFOR	Foreshadowed	I.iii	16
RATFOR-86	RATional FORtran	I. iv	6
RATFOR-M4	RATional FORtran	I. iv	6
RS NATIONAL PTS	Q & A re difficulties obtaining manuals	I. ii	86
RS-232	Accessing the RS-232 port from BASIC	I. iv	39
RSCC	Pat on the back and how to send up a friend	I. iv	14
RSHARD	ARCHIVE/RESTORE	I. iv	23
RSHARD	RSHARD from JCL	I. iv	23
RSHARD	Utility to support HD drivers use with DOS x.3	I.iii	7
SAID	Editors: SAID	I. i	36
SAID	Problem with replacing double occurring characters	I.iii	98
SAID	Product Highlights	I. ii	81
SAID	Upgrade to 1.1	I. ii	7
SCRIPSIT	Previewing output	I. iv	62
SET2RAM	SET2RAM and MEMDISK driver	I. iv	21
SETBAUD	by Jeff Brenton - change to program	I. ii	7
SIDEWAYS	SIDEWAYS with MC	I. iv	88
SIG	General Info on LDOS/TRSDOS6 Special Interest Grp.	I. i	6

SOFTWARE	Economic factors dictating pricing and continuity	I.iii 11
STRIP/BAS	by James Beard, Ph.D. - Strip last byte of a file	I. iv 59
TANDON	Bigger bubbles for your 5-megger	I. i 86
TANDY	1000 EX, SX, HL	I.iii 105
TANDY 2000	Q & A	I. iv 47
THE GUIDE	Mistake on page A-183	I. i 11
THE GUIDE	The Programmers Guide to LDOS /TRSDOS VERSION 6	I. iv 2
THE GUIDE	What happened to THE GUIDE?	I. i 2
THE SOURCE		I. iv 16
THE SOURCE	No reprint	I. ii 3
THE SOURCE	THE SOURCE to TRSDOS 6.2	I.iii 98
TMQ	TMQ kudos	I. iv 15
TRSDOS	Accessing the RS232 port from BASIC	I. iv 39
TRSDOS	Caution for Directory Tinkerers	I. i 14
TRSDOS	DEBUG Help	I. iv 39
TRSDOS 6	ECI's: Hints and Kinks	I. i 12
TRSDOS 6	Q & A	I. ii 24
TRSDOS 6.2	Add @VDPRT SVC to TRSDOS 6.2	I. i 15
TRSDOS 6.2	Caution about using unauthorised patches from BBSs	I. iv 45
TRSDOS 6.2	Problem with reverse video when exiting HELP	I. ii 67
TYPEIN	LS Utility. Q & A	I. ii 82
TYPEIN	TYPEIN & Profile 4+	I.iii 98
UNDATE	by Luis M. Garcia-Borro	I. iv 56
UNREL	REL to ASM translator	I. ii 11
UNREMOVE	Recovering REMOVEd files	I. iv 18
VIDTEX	Downloading	I.iii 33
VIDTEX PLUS	Patch to change default baud rate to 1200	I.iii 32
X-FTS	General info	I. i 9
X-FTS	Q & A	I.iii 99
XLR8	General bits & pieces	I. ii 86
XLR8	Notes on the H.I.Tech XLR8	I. ii 89
XLR8	SET2RAM and MEMDISK driver	I. iv 21
XLR8	XLR8er "Ramdisk"	I.iii 109
XMODEM	General info	I. i 9
XMODEM	Q & A	I.iii 32
Z-TIME1	Z-TIME1 Clock Board	I.iii 104
ZAP	Means of getting patches	I.iii 26
ZCAT	Product Highlight	I. i 76
ZSHELL	Product Highlight	I. i 76
ZSHELL	Product Highlights	I. ii 83

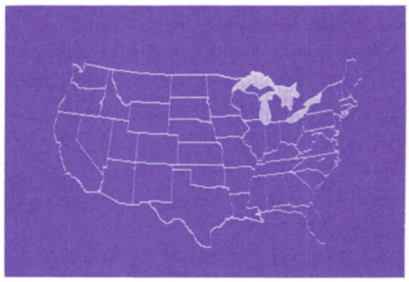
Grafyx Solution™ Save $170.00

Hi-Resolution Graphics for Mod 4/4D/4P/III



Superior Hardware. The Grafyx Solution provides 153,600 pixel elements which are arranged in a 640 × 240 or on the Model III a 512 × 192 matrix. Hundreds of new business, personal, engineering, and educational applications are now possible. The hi-res display can be shown on top of the standard display containing text, special characters, and block graphics. This simplifies program debugging, text labeling, and upgrading current programs to use graphics. The Grafyx Solution fits completely within any tape or disk based Model 4, 4D, 4P, or III. Installation is easy with the plug-in, clip-on Grafyx Solution board.

Superior Basic. Over 20 commands are added to the Basic language. These commands will set, clear or complement points, lines, boxes, circles, ellipses, or arcs. The hi-res screen can be printed on any of 30 popular printers or saved or loaded to disk without leaving Basic. Areas may be filled in with any of 256 patterns. Sections of the screen may be saved and then put back using any of five logical functions. Labels can be printed in any direction. The viewing area can be changed. The entire screen can be complemented or cleared. Graphics Basic provides dot densities of 640 × 240, 320 × 240, 160 × 240, and 160 × 120, all of which can be used in the same display.



Superior Software. The board comes with over 40 programs and files which make it easier to use, serve as practical applications, demonstrate its capabilities, and serve as programming examples. The software works with TRSDOS 1.3, 6.1.2, 6.2, 6.3; Dosplus 3.4, 3.5, 4; LDOS; and Newdos80. The Grafyx Solution is also supported by 30 optional applications programs: Draw, Bizgraph, xT.CAD, 3D-Plot, Slideshow, Mathplot, Surface Plot, Chess, etc.

The Grafyx Solution package is shipped complete for $129.95 (reduced from $299.95). The manual only is $10. Payment may be by check, Visa/MC, or COD. Domestic shipping is free on pre-paid orders. Texas residents add 7% tax.

MICRO-LABS, INC. 214-235-0915
902 Pinecrest, Richardson, Texas 75080

Graphics Solutions

High-Resolution Software and Hardware

GBASIC 3.0 - Radio Shack Model 4/4D/4P/III hi-res board owners take note of an enhanced graphics Basic: GBASIC 3.0 It not only provides an equivalent for each of the BASICG commands but adds a number of important new ones while using less memory. Without having to exit Basic, the hi-res screen can be saved to disk, loaded from disk, or printed on any of 30 popular printers: Epson, Star Micronics, Radio Shack, Okidata, C. Itoh, NEC, etc. The software works with TRSDOS 1.3, 6.1.2, 6.2, 6.3; Dosplus 3.4, 3.5, 4; LDOS; and NEWDOS80. The disk contains 40 graphics programs/files. Also included is a detailed manual with assembly language entry addresses. $39.95. (Specify Model 4 or III mode or add $10 for both.)

The following eleven programs run on a Model 4/4D/4P/III equipped with a Radio Shack graphics board and GBASIC 3.0 or a Micro-Labs Grafyx Solution board:

DRAW - A powerful full screen graphics drawing and editing program. $34.95.

BIZGRAPH - Create business graphs from hand-entered or VisiCalc data. $59.95.

xT.CAD - Professional drafting aid which outputs to a printer or plotter. $145.00.

SURFACE PLOT - Plot three-dimensional equations of the form Z=F(x,y). $39.95.

3D-PLOT - View three-dimensional data from any perspective or angle. $29.95.

MATHPLOT - Plot equations of the form Y=F(x) with auto scaling. $29.95.

CHESS - A very powerful program with 10 skill levels, 40 play options. $39.95.

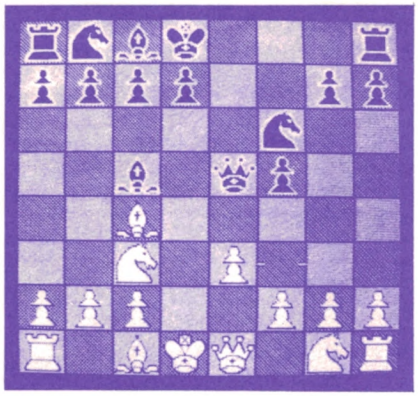
REVERSI - Play Othello with 10 skill levels, 20 execution options. $29.95.

3D Tic-Tac-Toe - Play the computer or a friend on a 4 × 4 × 4 matrix. $19.95.

SLIDESHOW - Create a sequence of hi-resolution picture displays. $19.95.

Biorhythm/USA - Plot your biorhythm or learn the states and capitols. $14.95.

JOY-MOUSE - Allows a Radio Shack Color Computer joystick, mouse, or touch pad to be connected to any Model 4/4D/4P/III. Hardware provides X, Y position values from 0 to 255. $119.95.



GRAFYX SOLUTION - A plug-in, clip-on board enhances any Model 4/4D 4P/III to provide 640 × 240 dot graphics. (512 × 192 on a Model III) The board comes with a 56 page manual and a disk containing both model 3 and 4 mode versions of over 40 programs and files including GBASIC 3.0 which adds over 20 graphics commands to Basic. $129.95.

Please specify your exact system configuration when ordering or requesting information. Payment may be by check, Visa, Mastercard, or COD. Domestic shipping is free on pre-paid orders. Texas residents add 7% sales tax.

MICRO-LABS, INC. 214-235-0915
902 Pinecrest, Richardson, Texas 75080

MISOSYS

SOFTWARE FOR YOUR P.C.

Order Now 800-MISOSYS (647-6797)

ED/ASM-86™

- 80X86 Assembly Language development

An integrated editor, assembler, linker, and debugger for 8086, 80186, 80286, and 8087 assembly and disassembly. All features are in a single EXE file.

- a full-featured line EDITOR with intra-line editing, block move and copy, partial save and load.
- a full-screen turbo-type editor is supported on CGA-equipped systems.
- the ASSEMBLER supports structured coding; can directly generate a COM or EXE file; assembly to memory for faster debugging.
- The DEBUGGER includes symbolic disassembly of arbitrary code to disk.

ED/ASM-86 [M-86-030] . . ~~$124.95~~ + $5 S&H

Now only $99.95

RATFOR-86™

- Rational FORTRAN translator

A professional implementation of RATional FORtran by James Beard, Ph.D. Provides structure and greater portability to FORTRAN programs to reduce your programming time and effort dramatically over that required when writing directly in FORTRAN. Includes tutorial reference manual.

- compiler has extensive macro support
- "arith" macro for binary arithmetic
- short form READ and PRINT
- enhanced switch case support
- Requires a FORTRAN compiler

RATFOR-86 [M-86-73] . . . $99.95 + $5 S&H

MISOSYS, INC.
PO Box 239
Sterling, VA 22170-0239
703-450-4181 MC, VISA, CHOICE

VA residents add 4½% sales tax.
S&H: Canada add $1;
Foreign use S&H times 3

MISOSYS

SOFTWARE FOR YOUR P.C.

Order Now 800-MISOSYS (647-6797)

LB86™

- A useful and flexible data manager

Easily used by anyone for managing their data. It's menu driven for ease of use; absolutely no programming needed. Why settle for expensive overkill with data managers you have to program when economically priced LB86 will do mailing labels, customer lists, registration data.

- Store up to 65,534 records per data base
- Up to 1,024 characters per record
- Up to 64 fields per record
- Up to 254 characters per field
- Nine field types for flexibility
- Select and sort on up to 8 fields
- Keep 5 different indexes for data access
- 10 input/update screens per data base
- 10 printout formats per data base
- Extensive on-line help always available

LB86 [L-86-510] . . . . $74.95 + $5 S&H

DED86™

- A utility disk that won't leave you puzzled.

A powerful sector-oriented disk/file editor and a page oriented memory editor. Search your entire disk while you tag sectors or clusters to be "kept"; direct DED86 to write the sectors as a disk file. Great for file recovery without fussing over FATs. DED86 gives you important features:

- Position by cylinder/head/sector
- Position by logical sector or cluster
- Jump about subdirectories
- Keep sector/clusters/memory
- Edit bytes in hexadecimal or ASCII
- Search strings in ASCII or hexadecimal
- DOS subshell available for your use

DED86 [M-86-020] . . $59.95 + $5 S&H

MISOSYS, Inc.
PO Box 239
Sterling, VA 22170-0239
703-450-4181 MC, VISA, CHOICE

VA residents add 4½% sales tax.
S&H: Canada add $1;
Foreign use S&H times 3